UNIVERSITY OF CANTERBURY

DOCTORAL THESIS

Using Mixed Reality for Asymmetric Remote Collaboration in a Room-scale Workspace

Author:
Lei GAO

Supervisor:
Prof. Robert W. LINDEMAN
Prof. Mark BILLINGHURST

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

The Human Interface Technology Laboratory New Zealand

August 31, 2020

Declaration of Authorship

I, Lei GAO, declare that this thesis titled, "Using Mixed Reality for Asymmetric Remote Collaboration in a Room-scale Workspace" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Lei Gao

Date: August 31, 2020

University of Canterbury

Abstract

Doctor of Philosophy

Using Mixed Reality for Asymmetric Remote Collaboration in a Room-scale Workspace

by Lei GAO

The primary goal of this thesis is to use Mixed Reality (MR) technology to enhance remote collaboration in a room-scale workspace. One of the fundamental requirements for remote collaborative system design is to help the remote expert to correctly understand the local worker's surrounding environment and support efficient remote communication, especially while working in a room-scale environment. To address this issue, I discuss the advantages and limitations of enabling a remote team to share a 3D view of the same workspace at the same time from different physical locations. I develop several prototype designs, and evaluate these designs with users on common collaborative tasks.

In this thesis, I present empirical results from five user studies on how Augmented Reality (AR) and Virtual Reality (VR) combined with 3D capture hardware, co-presence techniques, and efficient guidance cues can enhance the task performance and user experience of room-scale remote collaboration. To compare different interface design approaches, I developed a testbed that combines a low-resolution static 3D point cloud capture of the environment surrounding the local worker with a high-resolution real-time view of small focused details. User studies with the system found that the use of a 3D virtual representation can effectively improve the remote expert's spatial awareness of the local work environment. I also found that a high-resolution local view is always helpful for guiding, no matter if it is 2D or 3D, especially for complex operations. Furthermore, mutual awareness is an important factor in supporting natural communication.

This dissertation contributes to a more comprehensive understanding of MR remote collaboration systems' interface design in various guiding scenarios. As a result, my research explored some basic design principles of an MR remote collaboration system while working in a room-scale workspace and pointed out future research directions.

Acknowledgements

I have been lucky enough to have a chance to spend a few years of my life holding this research. The work presented in this thesis would not have been possible without the help of many people. I would like to acknowledge and thank them for their support.

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Robert W. Lindeman for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I would also thank my co-supervisor Prof. Mark Billinghurst. It has been an honor to be his Ph.D. student. I appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience productive and stimulating. I also thank Dr. Huidong Bai for giving encouragement and sharing insightful suggestions. His endless guidance is hard to forget throughout my life.

It was a pleasant and fruitful working experience at the HIT Lab NZ. I would like to thank the staff and students there for their support to make this Ph.D. thesis possible. I would also like to extend thanks to all of the participants of my studies who helped me get results of better quality.

Last but not least, I would like to thank my family. To my parents, thank you for supporting me spiritually throughout my life.

Contents

List of Figures

List of Tables

List of Abbreviations

AR	**A**ugmented **R**eality
VR	**V**irtual **R**eality
MR	**M**ixed **R**eality
HMD	**H**ead-**M**ounted **D**isplay
HHD	**H**and-**H**eld **D**evice
FOV	**F**ield **O**f **V**iew
POV	**P**oint **O**f **V**iew
DOF	**D**egrees **O**f **F**reedom
FPS	**F**rames **P**er **S**econd
SLAM	**S**imultaneous **L**ocalization **A**nd **M**apping
ICP	**I**terated **C**loset **P**oint Algorithm
CVEs	**C**ollaborative **V**irtual **E**nvironments

Dedicated to my dear parents.
Thank you for all of your support along the way.

Chapter 1

Introduction

1.1 General Overview

This Ph.D. thesis explores how Mixed Reality (MR) technology can enhance remote collaboration in a room-scale workspace. Current worldwide fast data connections and the Internet can enable a remote expert to see a local worker's physical workspace in real-time and help them effectively perform a task. In this case, remote collaboration becomes more accessible and can almost be as effective as face-to-face communication.

A typical remote collaboration system connects a local workspace where a worker needs help with an unfamiliar physical task to a remote expert who can provide guidance via communication cues such as speech, pointing, or virtual annotations. For example, remote Augmented Reality (AR) software [61] allowed a local worker to share a video of their workspace with a remote expert who can place virtual annotations onto the video and share it back so the local worker can see the annotations (Figure 1.1). In systems like this, 2D video is often used to show the remote expert a view of the local worker's current workspace. However, without a truly three-dimensional view of the local scene, the remote expert may not be able to correctly understand the spatial relationships between objects in the local environment.

Rendering a 3D capture of the local workspace in immersive Virtual Reality (VR) provides a natural way to help remote experts gain a better spatial understanding of the local working environment. However, it separates the experts from the real world and restricts the guidance cues [8]. One alternative approach is through MR, which seamlessly combines the "virtual space" and "reality" together in the same visual display environment [83]. This enables users to see each other and the real world simultaneously and facilitate effective communication and intuitive manipulation of the target objects.

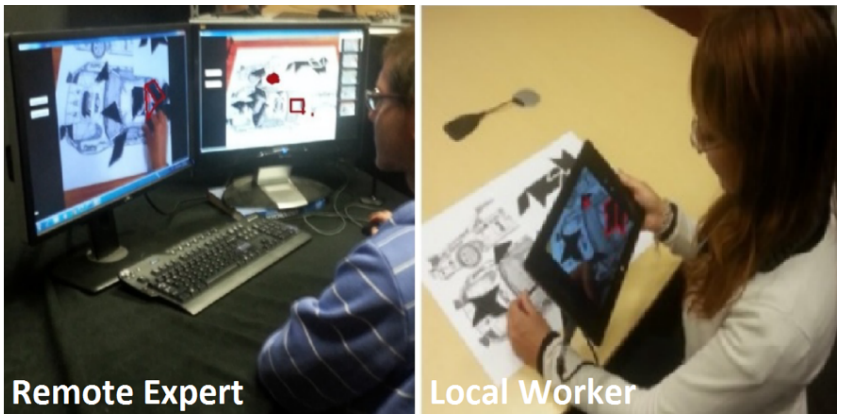


FIGURE 1.1: An example of remote collaboration system [61]

To provide accurate and efficient help, the remote expert needs to first **learn the spatial arrangement** of the local environment, and then **search for an object of interest** and **direct the local worker to locate** and **operate** it. During this process, there are several key questions, including:

- How can a remote expert learn the spatial distribution of the items in the local environment?
- Does the remote expert need an overview of the task space, or only the local worker's first-person view?
- How does the remote expert behave when he or she tries to understand the physical layout of the local environment?
- What type of interface would users on both sides like to use while working on a task?
- How should the remote expert communicate with the local worker?
- How should the remote expert guide the local worker to move to the target location and operate the target, by pointing or speech?

The answers to these questions could be beneficial in a number of ways. First of all, they suggest efficient ways to present the local view to the remote expert. Secondly, they help integrate different types of interfaces to complete a set of tasks for different purposes. Last but not least, they show possible solutions for improving communication cues between the local and remote users. In particular, they provide the basis for enhanced interface design for Mixed Reality (MR) remote collaboration systems. In this Ph.D. research, I will analyze these issues based on the results from five different user studies.

This research explores new approaches for MR based remote collaboration in a room-scale workspace on top of a local scene capture and sharing system. The system allowed users to perform the necessary collaborative tasks in the ways as if they were face-to-face, or even the ways not possible while working face-to-face.

1.2 Example Scenarios

There are many possible examples of how a room-scale remote collaboration system could be used. Figure 1.2 shows the sketch of a scenario where a person is getting some advice on home furnishings from a remote expert. The local user (on the right) walks through their house wearing an AR see-through head-mounted display (HMD) with 3D scene capture hardware mounted on it. The captured 3D scene is streamed to a remote expert (on the left) who is wearing a VR display and can freely move through the VR version of the local user's home. The local user can see the expert as a virtual person in their real world. The remote expert's hands, head motion, and eye gaze are tracked so that the users can exchange rich communication cues. Working together, the remote expert can provide advice for the local user on how to redecorate their home. For example, the remote expert can place objects in the virtual copy of the local user's real space and have these appear as AR objects in the local user's real view, showing how their room could look like when furniture is added. Similarly, the remote expert can point with their virtual hands to objects in the real world, or draw virtual annotations to explain their ideas.

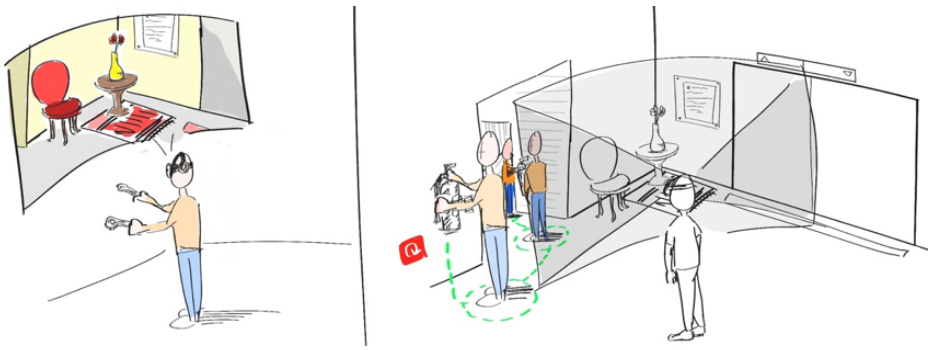
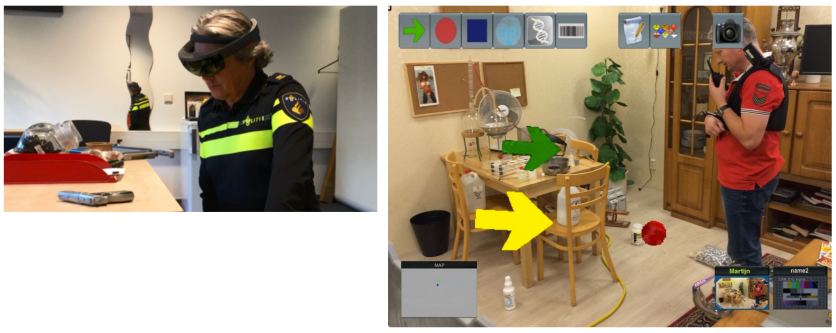


FIGURE 1.2: MR remote collaboration for interior design

In another example, when a crime occurs, crime scene investigators need to inspect the scene as quickly as possible. However, there is often an expert in a particular area (e.g., blood splatter analysis) that should be there but is in a remote office many hours away. Previous research has shown how a wearable computer with a head-mounted display and AR interface can be used to allow investigators at the crime scene to share the view and get remote expert help [22] [23] [95] (Figure 1.3). In this case, the researchers

used SLAM tracking to allow the remote expert to add AR cues indicating points of interest in the investigators' view of the crime scene. Our research will move beyond this, and using our technology, a 3D environment of the crime scene could be captured and shared, enabling the remote expert to feel like they are in the scene itself and can collaborate more effectively.



Investigator Using an AR Display *The AR View with Remote Expert Annotations*

FIGURE 1.3: Using AR for remote crime scene investigation [22] [23]

In both of these scenarios, in order for the remote expert to assist the local person, there is a need for room-scale tracking and the ability to capture and share 2D imagery, video, and 3D geometry. In addition, the use of shared hand gestures, virtual viewpoint representation, and shared annotations will allow for a better understanding of the focus of attention and the sharing of rich communication cues.

1.3 Problem Statement

In order to achieve effective remote collaboration, the local workspace needs to be shared with a remote expert using audio, 2D video, or 3D scene cues. Compared with audio-only communication, sharing visual communication cues enable remote collaborators to ground their utterances more quickly and establish better shared understanding [95]. However, there are still a large number of challenging research questions that need to be addressed while bringing the remote collaboration to a large-scale workspace.

The most common way to share the local view is using either a video camera with a fixed position [95] [64] or a dynamic moving video camera [64] [46] [97]. However, these approaches either limited the size of the local workspace as the camera's field of view (FOV), or could be too shaky for the remote expert to watch comfortably for a long time [3]. For room-scale remote collaboration, the entire local physical geometric layout is required to be shared with the remote expert to enable him/her to virtually walk

through the room and experience the physical space from a range of perspectives. In this case, independent viewpoint control and 3D scene sharing are required.

To present the spatial layout of the local workspace, the depth sensor has been used to capture the local scene. In typical remote collaboration systems [112] [107] [1] with 3D scene sharing enabled, the view from this kind of setup was displayed on 2D monitors which still separated the experts from the 3D local representation. Recent research [4] [72] showed the possibility of using HMDs to help the remote expert immerse himself/herself in a 3D virtual copy of the local worker's space. However, compared to the 2D video, the resolution (usually 640 * 480 for the depth image used for 3D modelling) of the 3D models was pretty low at this moment, so these real-time 3D reconstructions might not yet be suitable for detailed manipulation.

Besides viewing a 3D replica of the local scene, it is critical for the remote expert to accurately perceive local worker's reactions with a high level of awareness during a collaborative task process [121]. Traditional remote collaboration setups focused on small workspaces that could be captured in one view. Therefore, the worker's actions and local changes were straightforward for the expert to observe from the shared view. In contrast, the worker could easily move out of the expert's current view while the shared scene is room-sized with free movement enabled. The expert would better control the task process if he could quickly locate the partner's location and the local changes. On the other hand, the worker also required the approach to efficiently locate the remote guiding cues in the large space to enhance collaborative performance.

Hand gesture [4] showed the potential of providing guiding cues most naturally, but it has been limited by the capturing device. Sensors used for gesture detection only support narrow FOV and short detection range, which work well for small workspace with the operating area just in front of the user. However, room-scale workspace usually contains more complicated task situations, such as the expert may need to indicate several targets spread all over the workspace at one time. In this case, the hand gestures restricted in a small focused area are not efficient anymore. There is a need to design powerful guiding cues for room-scale remote collaboration.

Overall, the remote collaboration system design in room-scale workspace includes the following key characteristics compared to traditional remote collaboration system:

- Capture and share the entire local physical geometric layout to enable the remote expert freely move and experience the space from different perspectives;

- Show detailed local view upon the 3D scene to overcome current hardware and software limitations in 3D capturing;
- Enable the expert to catch the local worker's actions and the local changes quickly and efficiently;
- Enable the worker to catch the remote guiding cues efficiently;
- Support larger-scale capabilities for the guiding cues;

Previous research in the field of remote collaboration was limited in a small working area supported by either the 2D video with less depth information [3] [64] [71] or a 3D representation with low resolution for viewing [112] [107]. In this research, I presented an MR remote collaboration system with a hybrid viewing interface for the remote expert. It combined a low-resolution 3D scene of the environment surrounding the local worker with a high-resolution real-time view of small focused details in a room-scale workspace. The remote expert could see a virtual copy of the local workspace with independent viewpoint control. Meanwhile, the expert could also check the local worker's current actions through a real-time feedback view. Furthermore, guiding cues and virtual view frustums were also supported to improve the mutual awareness between collaborators.

1.4 Research Questions

In this Ph.D. thesis, I focused on studying remote collaboration in a room-scale workspace by using MR. The overall research objective was to explore the following research questions based on usability evaluation and collaborative experience analysis through five user studies with different interface designs:

Q1 : Can AR and VR combined with 3D capture hardware, co-presence techniques, and efficient guidance cues enhance the task performance of room-scale remote collaboration?

Q2 : Can AR and VR combined with 3D capture hardware, co-presence techniques, and efficient guidance cues enhance the user experience (e.g., usability, social presence, and motion sickness) of room-scale remote collaboration?

Q3 : Would the combination of different remote collaboration media (3D, 2D, and 360° panorama) complement each other for room-scale collaboration?

Q4 : Would the users' behaviors provide some hints on exploring the guidelines for room-scale remote collaboration system design?

Based on the above research objective and questions, I hypothesized that MR based remote collaboration systems could support better task performance and user experience for room-scale collaborative tasks than traditional video or simple 3D sharing technologies. I also asserted that different remote collaboration media could complement each other if we efficiently combine them. I measured this based on usability evaluation and user experience analysis with different user interface designs. Furthermore, users may behave differently during each task step, which may reveal some guidelines for room-scale remote collaboration system design.

In this thesis, I tested my hypothesis based on following steps. I first enhanced the local scene capturing from 2D to 3D with independent viewpoint control, then enlarged the workspace to room-sized for evaluation. In the next step, I introduced different combinations of techniques to support remote collaboration in a large space. I finally analyzed the users' behaviors during the remote collaborative process to summarize basic interface design principles.

1.5 Research Contributions

This is one of the first Ph.D. theses that explores the use of MR interfaces and rich environmental cues for room-scale remote collaboration on physical tasks. The research focused on system development, user experience prototyping, and evaluation studies, not on creating low-level technology such as point cloud stitching, position tracking, or gesture capture. Many of these elements were already available from commercial or research sources, so I explored how to combine them in new and exciting ways to support remote collaboration.

To support large-scale remote collaboration, I developed a prototype system that reconstructed the local physical environment and shared it as a 3D VR static scene with a remote expert. Based on the HMD position tracking, the expert could freely navigate himself/herself through this 3D virtual copy to better understand the spatial relationship between objects in the large local workspace. In this case, the remote experts might feel they were sharing the same workspace as the local workers.

Our remote collaboration system captured and reconstructed the local scene as a static 3D scene. Once created, there was no real-time update of the 3D scene from the local

worker's side. However, I developed different interface ideas to provide real-time feedback to show the local worker's actions. I also conducted five user studies to evaluate these different interface designs.

In the first user study (Chapter 3), I reported on a prototype system for remote collaboration using VR headsets with an external depth camera attached. This system wirelessly shared not only the full 3D captured environment data but also real-time orientation info of the worker's viewpoint. It could enhance the remote expert's understanding of the local spatial layout. However, it also increased the remote expert physical stress during the task process.

In the second user study (Chapter 5), I extended the MR system to support the capture of the entire local physical work environment. By integrating the keyframes captured with the external depth sensor into one single 3D point cloud data set, the system could reconstruct the entire local physical workspace into the VR world. In this case, the remote expert could observe the local scene independently from the local worker's current head and camera position, and provide gesture guiding information even before the local worker was looking at the target object.

In the third user study (Chapter 6), I presented an MR system with a combination of different view media to support remote collaboration. By combining a low-resolution 3D point cloud of the environment surrounding the local worker with a high-resolution real-time view of small focused details, the remote expert could see a virtual copy of the local workspace with independent viewpoint control. Meanwhile, the expert could also check the current actions of the local worker through a real-time feedback view.

In the last two research studies (Chapter 7), I investigated how users behaved during the remote collaborative process while using an MR interface, especially for remote experts. I found that the remote expert preferred to learn the local physical layout and search for the targets with a global perspective from the 3D static view. The results also showed that the expert chose to use the 360° live view with independent view control rather than the 2D first-person view with high-resolution imagery to control the task procedures and check the local worker's actions.

Overall, my research had the following novel aspects and contributions:

- A novel remote collaboration system that combined AR, VR and 3D space capture (Chapter 4);

- A 6 degree of freedom (DOF) representation of both the remote expert and local worker in the shared 3D virtual space to enhance spatial awareness and mutual awareness (Chapter 4);
- High-resolution real-time local feedback together with a low-resolution 3D reconstruction of the room-size workspace to help the remote expert to learn the local situation and control task process (Chapter 6, 7);
- A group of visual cues that enabled users to effectively communicate with each other under complicated task conditions in the room-scale workspace (Chapter 5, 6, 7);
- Five corresponding user studies to evaluate interface design principles for remote collaboration systems while working in a room-scale workspace (Chapter 3, 5, 6, 7);

In summary, the results presented in this thesis revealed how MR could be effectively utilized in a room-scale remote collaboration system design. Furthermore, the user study design and evaluation processes presented in this thesis explore possible methods for designing and evaluating MR based interfaces for remote collaboration in the room-scale workspace.

1.6 Thesis overview and Outcomes

In the next Chapter (Chapter 2), I provide a brief literature review that covers 2D/3D interface and communication cue design for remote collaboration systems, 2D/3D environment capture techniques for MR based systems, and mutual awareness evaluation in the field of multi-user based VR system designs.

In Chapter 3, I present an MR remote collaboration system that enables 3D scene capture of the local workspace with real-time updates to show local changes. Based on this research, I try to bring the remote collaboration from 2D video streaming to 3D view sharing.

In Chapter 4, I introduce the implementation of our static scene capture and sharing system. Using this system setup, I can share the room-scale local workspace as a dense point cloud with a remote expert. In Chapter 5 to Chapter 7, I report on four user studies conducted to evaluate the interface design and analyze user behavior while using our collaborative system.

The system I introduce in Chapter 4 used VR headsets connected to powerful PCs;

therefore, it could only be tested in an indoor experiment environment but not in an outdoor working scenario. In Chapter 8, I present a new remote collaborative system setup based on mobile devices that could be used anywhere and anytime.

The list of publications below gives an overview of the scientific activities and the collaborations which occurred during the work of this thesis. The following publications provide the main contribution of this thesis. The author was the main contributor at all stages of the work – forming the design ideas, designing and implementing the interfaces, designing and conducting user evaluations, and analyzing and discussing the results.

- Lei Gao, Huidong Bai, Gun Lee, and Mark Billinghurst. "An oriented point-cloud view for MR remote collaboration". In: *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications*. ACM. 2016, 8:1–8:4

- Lei Gao, Huidong Bai, Robert W Lindeman, and Mark Billinghurst. "Static local environment capturing and sharing for MR remote collaboration". In: *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*. ACM. 2017, p. 17

- Lei Gao, Huidong Bai, Thammathip Piumsomboon, Gun A Lee, Robert W Lindeman, and Mark Billinghurst. "Real-time visual representations for mixed reality remote collaboration". In: *Proceedings of the 27th International Conference on Artificial Reality and Telexistence and 22nd Eurographics Symposium on Virtual Environments*. Eurographics Association. 2017, pp. 87–95

- Lei Gao, Huidong Bai, Mark Billinghurst, and Robert W Lindeman. "User Behaviour Analysis of Mixed Reality Remote Collaboration with a Hybrid View Interface". In: *32nd Australian Conference on Human-Computer Interaction*. 2020, pp. 629–638

- Lei Gao, Huidong Bai, Weiping He, Mark Billinghurst, and Robert W Lindeman. "Real-time visual representations for mobile mixed reality remote collaboration". In: *SIGGRAPH Asia 2018 Virtual & Augmented Reality*. ACM. 2018, p. 15

In the following publications, the author was mostly involved in the interface design and system implementation, making minimal contributions to the evaluations of the studies and data analysis.

- Huidong Bai, Lei Gao, and Mark Billinghurst. "6DoF input for hololens using vive controller". In: *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*.

ACM. 2017, p. 4

- Yuanjie Wu, Lei Gao, Simon Hoermann, and Robert W Lindeman. "Towards Robust 3D Skeleton Tracking Using Data Fusion from Multiple Depth Sensors". In: *2018 10th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)* (2018), pp. 1–4

- Prasanth Sasikumar, Lei Gao, Huidong Bai, and Mark Billinghurst. "Wearable RemoteFusion: A Mixed Reality Remote Collaboration System with Local Eye Gaze and Remote Hand Gesture Sharing". In: *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE. 2019, pp. 393–394

Chapter 2

Related Work

In this section, I first focused on reviewing previous research in the field of remote collaboration in terms of both interface design and communication cues. Then I also provided a brief review of techniques for environment capture and ideas for presenting mutual awareness in VR.

2.1 Remote Collaboration Interfaces

The basic idea of remote collaboration interface design is to create a system that shares the current local situation with a remote person using audio, 2D video, or 3D representation cues. There are many possible types of remote collaborative systems. In my research, I mainly focused on asymmetric remote collaborative tasks requiring the local worker's physical actions under the guidance of a remote expert. For example, a local worker fixes a complex machine following a remote engineer's instructions over a video link. This is an example of asymmetric collaboration because the two participants have different roles, with the information flow mostly from the remote expert to the local worker.

2.1.1 2D Interface

With the development of widespread fast data connections and the availability of high-end computing and communication devices, the potential for remote collaboration has dramatically increased [37]. Providing a remote expert with visual information through video of the local worker's workspace where the task is being performed is one method for remote collaboration [29] [3] [97].

Compared to face-to-face collaboration, video sharing systems restrict the sharing of communication cues by their limited field of view of the cameras; however, they still have more advantages than audio-only systems while completing collaborative tasks [29].

For example, when the task involves manipulating objects that are difficult for the users to verbally describe, a video of the shared workspace is more valuable than audio-only communication.

There was a wide variety of different video sharing systems that have been developed in the past to support remote collaboration. Figure 2.1 shows a typical remote collaboration system [3] that used overhead fixed video cameras on each side to combine the view of both workspaces together and displayed the result on monitors in front of the users. This setup was a static video-sharing system, because the cameras did not move, and was considered to be useful for monitoring the progress of tasks in a constrained workspace that did not require complicated manipulation, or where the workers did not move around a lot.

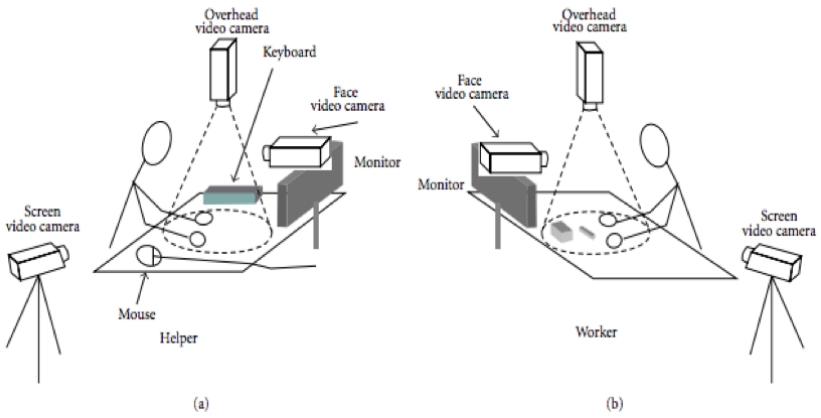


FIGURE 2.1: Technical setup of a typical remote collaboration system [3]

By using a cursor or pointing hand gestures, the system had the capacity to establish a joint focus of attention for the users. Using this technical setup, Alem et al. [3] presented a study exploring how using hand gestures or a cursor pointer affected collaboration. Remote experts were asked to use gestures or a cursor to help workers finish physical assembly tasks. Both conditions were similar in their assembly time and accuracy; however, hand gestures were richer than using a cursor in terms of representations of object rotation and orientation in the task. Furthermore, the participants felt that the hand gestures produced higher quality collaboration compared to the cursor condition.

Kirk et al. [64] designed a more natural gesture-based remote collaboration system to help a local worker understand guidance from a remote expert (Figure 2.2). A video camera was used to capture the movement of the remote expert's hands, and the gestures made were then projected onto the surface of the local workspace. Projecting the remote guidance directly on top of the local workspace provided a tangible interface for direct

manipulation of physical objects. Physical touch and manipulation played an essential role in interpersonal communication [12]; however, current GUI-based interface design separated the computer input from communication in the real world, in which case users could not directly interact with the physical targets. Therefore, tangible interface design, as shown in projects such as illuminating light [118], metaDESK [116], and mediaBlocks [117], could effectively increase the users' spatial and kinesthetic senses in understanding the environment. In this case, researchers found that projected gestures could provide a significant advantage for remote collaboration tasks during the early stages of an interaction. They also found that the task completion time and errors could be decreased while using voice and gesture together.

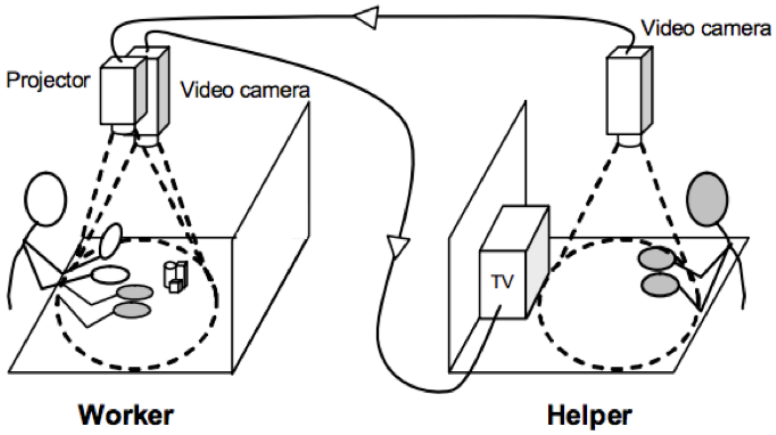


FIGURE 2.2: Projecting hand gestures into the physical world [64]

Conventional remote collaboration systems on physical tasks typically shared 2D video feed between local and remote users to help them understand their partner's situation [3] [64] [62]. This type of system often used fixed-view cameras, which limited the size of the viewing and work volumes for the remote user. Therefore, researchers had started to explore alternative approaches such as using head-mounted cameras [30], hand-held cameras [107] [6], or cutting between shots from multiple cameras [40] to support dynamic views from different positions and poses. Based on this changeable point of view, the remote expert could observe the local workspace from a series of directions, which improved the understanding of the worker's local situation [30].

Some studies showed that automatically following the local worker's actions could effectively reduce the remote expert's cognitive load; therefore, the camera should follow the local worker's point of view (POV). For example, if a worker wore a head-mounted camera to capture the view of the local workspace, the remote expert could share the same first-person POV as the worker and guide him/her during the completion of the

task [30] [61] (Figure 2.3, left). Another possible solution was to automatically track an indicator, such as the worker's hand, in the local work environment. In this case, the remote expert could quickly pay attention to what the local worker was currently working on [97] (Figure 2.3, right).

FIGURE 2.3: Head-mounted camera (left) [30] and using the camera to automatically track the worker's hand (right) [97]

A head-mounted camera could be used to share a local worker's first-person view with a remote expert, but it could still be difficult for the expert to see exactly where the worker was looking at in the shared view. Since gaze could convey information between users [67], researchers had started to use gaze tracking in remote collaboration tasks. Gupta et al. [46] used an eye-tracker attached to the head-mounted display to locate the exact point of gaze of the local worker (Figure 2.4, left). For the remote expert, a monitor showed the local worker's view with their eye gaze indicated on top of it (the red dot on Figure 2.4, right).

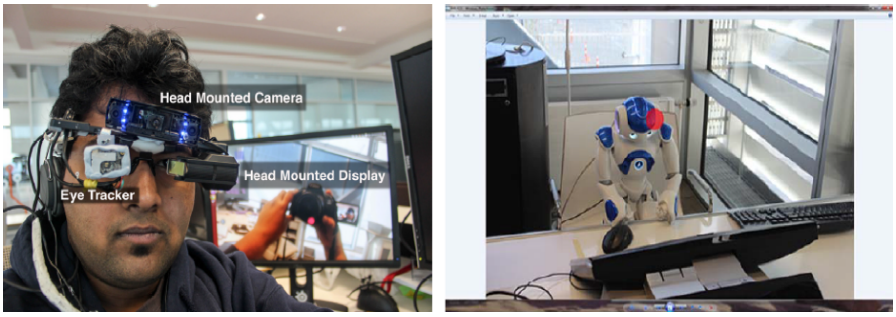


FIGURE 2.4: Gaze tracking hardware prototype for a local worker (left). The user interface for the remote expert (right) [46]

During their study, local workers were asked to construct LEGO models with and without the gaze tracking, assisted by a remote expert using a cursor pointer (Figure 2.5). The result showed that gaze tracking could be used to change the nature of remote collaboration with head-mounted systems. For example, the remote expert could be aware of the local worker's implicit intentions even before he/she physically started

interacting with the items in the workspace. This was because people always look at objects before they manipulate them. Furthermore, they found that providing gaze cues could significantly improve the remote collaboration performance even without supporting pointing guidance from the remote expert. However, other researchers found that interpretation of the user's communicative intention could be complicated while just using gaze tracking in remote collaboration [86]. Based on the research of Muller et al. [87], the uncertainty of gaze transfer also resulted in longer solution time and more verbal effort as users relied more strongly on speech communication. In this case, the best way to use gaze tracking in remote collaboration could be to combine it with other assistance cues, such as speech and hand gestures.

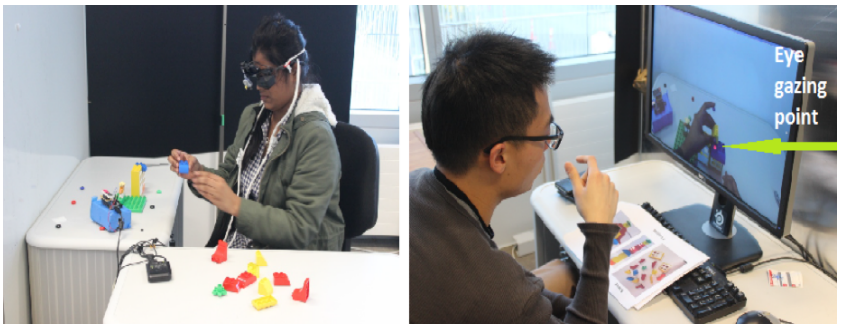


FIGURE 2.5: Local worker assembling LEGO (left); Remote expert guiding the local worker (right) [46]

Higuch et al. also explored how the remote expert's gaze could be used as a communication cue during remote collaborative tasks [49]. They pointed out that gaze movement could be used for predicting the intentions of the expert's next interest or instruction. However, they indicated that eye gaze could only be used for explicit instructions while combining it with speech. Other researchers also tried to support dual eye-tracking [20] [15] [54] during collaborative tasks, which simultaneously recorded the gaze of both the local worker and remote expert. The results of these research verified that sharing gaze information could improve pair performance in the field of collaborative conceptual learning and collaborative search [104]. However, gaze information could be a noisy communicative signal; therefore, it was important to understand how to represent gaze cues to best support remote collaboration [78].

Using a head-mounted camera to share the local view meant that the remote viewpoint was controlled by the local worker, which had some problems. The view captured by a head-mounted camera could be quite unstable for the remote expert; each time the worker moved his/her head, it would change the remote POV, which might interrupt the remote expert's awareness of the local workspace. The expert's view was limited to the same area that the worker saw while using the head-mounted camera, which

TABLE 2.1: Variables of 2*2 mixed user study design

Variable	Configuration
Within-subject: "Control"	Helper-control: control of the camera's point of view was fully driven by the expert
	Worker-control: control of the camera's point of view was fully driven by the worker
Between-subject: "worker's knowledge"	No knowledge: the worker has no background knowledge of the task solution
	Partial knowledge: the worker has partial background knowledge, but still needs help from a remote expert

also made it more difficult for the remote expert to understand the spatial relationship of items in the local physical workspace. Studies showed that, in some cases, a static wide-angle camera [30] or even 360° camera [58] [106] [73] could be used to enlarge the area captured in the local scene. In this case, the remote expert could independently control their perspective by viewing a specific part of the whole video frame or rotating around the 360° camera view. Therefore, viewpoint control had become one of the critical issues in the field of remote collaboration.

In order to investigate user performance and behavior related to the issue of who controlled the point of view in a remote assistance scenario, Lanir et al. [71] presented their research by using the 2x2 mixed user study design shown in Table 2.1. In their study, a camera was mounted in parallel with a laser pico-projector on the local worker's side, and used to capture the view of the local workspace, which was then transmitted to the remote expert's interface (Figure 2.6). Annotations created by the expert could be projected on the top of real objects in the local work environment by using the projector. During the worker-control condition, the worker controlled the device by manually moving it; on the other hand, during the helper-control condition, the expert remotely controlled the device using a robotic arm (Figure 2.7).

The user study tasks were carried out on a work desk 60x150 cm in size. Participants were asked to complete two different types of tasks: a LEGO construction task and a TV wiring task. The results showed that the impact of control of POV on user performance was significant but task dependent. They indicated that helper-control improved performance over worker-control while dealing with tasks that required more changes in

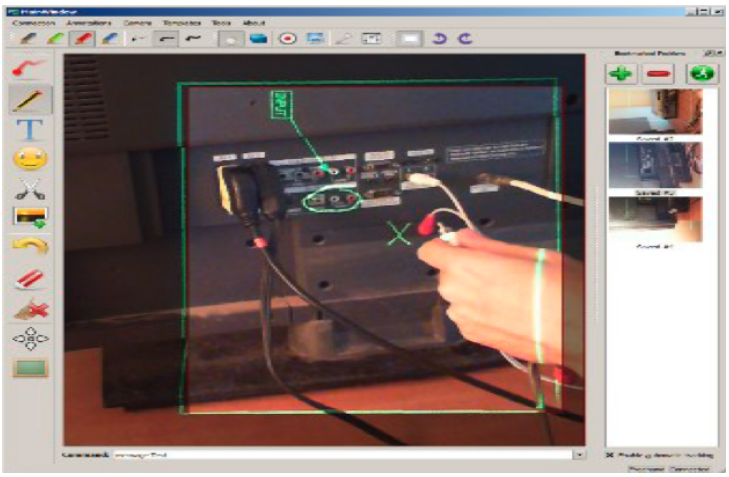


FIGURE 2.6: The expert's interface for the study of POV control [71]

FIGURE 2.7: Device for worker-control (left); Device for helper-control (right) [71]

the POV. In other words, helper-control was more beneficial in dynamic workspaces in remote collaborative tasks. Their study also showed that, during the same task, there were more changes in the POV in the helper-control condition than in the worker-control conditions. The experts did not share the same holistic workspace view as the workers, so in the helper-control condition, they always needed to move the POV around to follow the worker's actions in order to understand what the worker was doing.

Most of the current research in the field of remote collaboration tended to focus on improving the collaboration system to reproduce the face-to-face experience [39]. Therefore, how to increase the ability of remote collaboration systems to share experiences and support task-space conferencing became one of the major focus areas for researchers. To make users feel more connected to each other during remote collaborative tasks, some researchers tended to provide richer local context to remote experts, while others focused on using a virtual representation from the remote expert's side to improve the

feeling of being connected. Virtual representations, such as a pointer, annotations, or hand gestures, could significantly increase remote communication experience with 2D interfaces.

Drawing virtual annotations on live video from a head-worn or handheld camera might be challenging, since remote experts could easily lose their referents when the focus of view changed [47] [65] [70], such as the local user turning his/her head when wearing a head-worn camera. Spatial virtual references on physical objects provided an efficient cue for the local worker to understand the expert's guidance. Previous research either set a stationary camera [6] [31] [47] [18] or used extensive equipment [17] [90] to support world-stabilized annotations; however, these approaches restricted the size of the workspace or the movement of the local worker. One alternative was to use a simultaneous localization and mapping (SLAM) system for camera tracking [115]. SLAM systems could create a map of an unknown environment while simultaneously keeping track of the camera's location. In this case, the remote collaboration system could operate in environments of arbitrary geometric complexity, and support world-stabilized annotation and local virtual camera movements independently from the remote expert's viewpoint. This allowed the remote expert to place a virtual annotation on a real object, and have that annotation stay fixed with the object when the local worker changed their view (Figure 2.8).

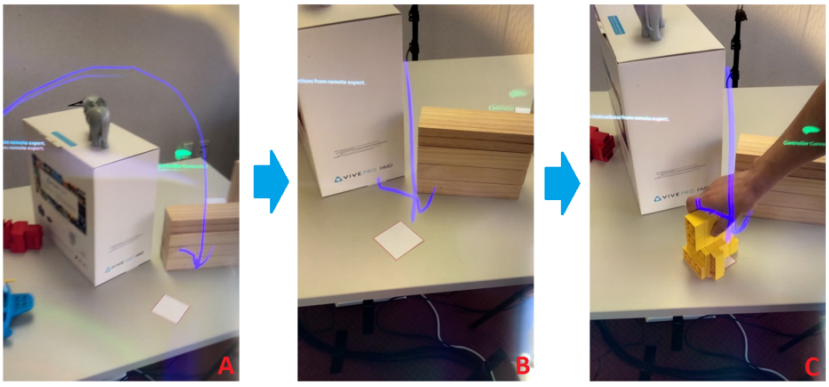


FIGURE 2.8: World-stabilized annotation to guide the local worker [101]

Kim et al. [61] developed a system for registering the visual communication cues provided by a remote expert into the real-world coordinate system, which could enable the local worker to see visual guidance cues in the 3D space. As shown in Figure 2.9, on the local worker's side, he/she could either use a head-mounted display or a handheld tablet with a USB camera attached to capture the first-person viewpoint. SLAM tracking was used to build a world coordinate system shared with the remote expert in an arbitrary physical scene without any prior knowledge. For the remote interface, the expert

could use a mouse to point or draw annotations on the shared video, and these virtual cues were sent back and displayed in the local worker's world coordinate system. In this case, the remote expert could guide the local worker on a physical puzzle assembly task.

FIGURE 2.9: Prototype system used by a remote expert (left) and a local user using a HMD (top right) or a Hand Held Device (HHD) (bottom right) [61]

Based on this system setup, they conducted a user study comparing three remote guidance conditions with different combinations of communication cues: (1) voice only, (2) voice + pointer, and (3) voice + annotation. The result showed that using a pointer was preferred by users over voice only and voice + annotation. However, using an annotation cue was also useful because it could remain visible in the scene.

Gauglitz et al. [38] [39] developed another system (Figure 2.10) based on SLAM tracking, and demonstrated in a scenario of remote assistance for car-engine repair. SLAM tracking was used to locate the position of the local worker's viewpoint and build a world-stabilized coordinate system in the 3D space. The system could project the local worker's current view onto the reconstructed engine model at the remote side and integrate visual symbols into a 3D position of the local workspace as augmented virtual clues. Both the local worker and remote expert's viewpoints could be tracked in the same shared world coordinate system; therefore, they could navigate their own view independently from each other.

2.1.2 3D Interface

Traditional 2D video sharing had the advantage of supporting a real-time high-resolution view for detailed manipulation, but could not easily represent the spatial layout of the captured scene, or provide depth information to the remote user. To overcome this

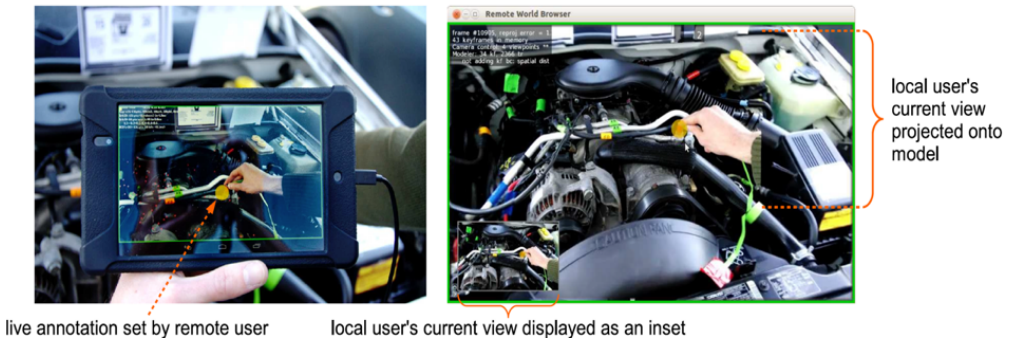


FIGURE 2.10: Car-engine repair remote collaboration interface: local worker interface (left); remote helper interface (right) [38]

limitation, researchers had also experimented with capturing the local workspace as a 3D geometry model [112] [107] [1]. This approach increased the remote expert's spatial awareness about the local environment, enabled independent viewpoint control, and addressed the occlusion issue (where objects in the foreground block others in the background).

Two possible ways had been used to present the captured 3D models. The first one was to display the 3D model on a traditional 2D screen, such as a computer monitor or a handheld device; the second was to use an HMD to show the view in a VR environment. A VR view was considered more immersive and more natural for viewpoint control. A study from Johnson et al. [55] showed that viewing on an HMD was advantageous for giving frequent directing commands during dynamic tasks.

Previous research had shown that supporting viewpoint independence increased spatial awareness in remote collaboration [71]. To explore this, Tecchia et al. [112] developed a prototype using an over-head depth sensor and HMD together. As shown in Figure 2.11, on the worker's side, a fixed over-head depth sensor was used to capture the physical work environment in 3D, and a monitor showed the view of the scene augmented with guiding information. On the remote expert's side, another depth sensor was used to capture the hand gestures performed by a remote expert, and an HMD was used to display the reconstructed 3D view based on the scene captured from the worker's workspace. Compared to sharing a 2D view from video cameras, the participants found that this 3D system allowed them to point to places and locations in the remote workspace that would not have been accessible before.

However, their approach had some distinct limitations. First of all, displaying remote guidance cues on a monitor required the worker to look at the monitor to check the feedback and take his/her eyes off the current task at hand, which disconnected the

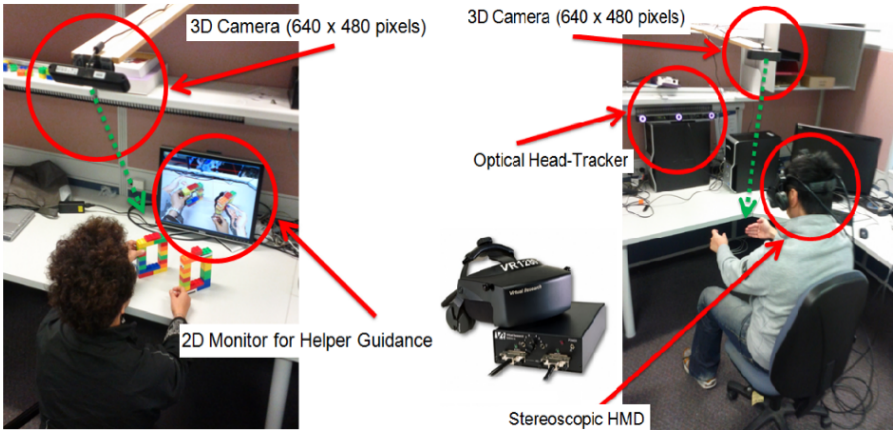


FIGURE 2.11: 3D helping hands system setup [112]

worker from what he/she was working on. Secondly, the top-view depth camera only captured the 3D scene from its own point of view, so it was subject to the occlusions caused by objects in the foreground. For example, if the expert moved his/her head too far away from the depth camera's view on the worker's side, gaps caused by missing data became obvious (Figure 2.12). In other words, a single static sensor could not support true 3D view independence.

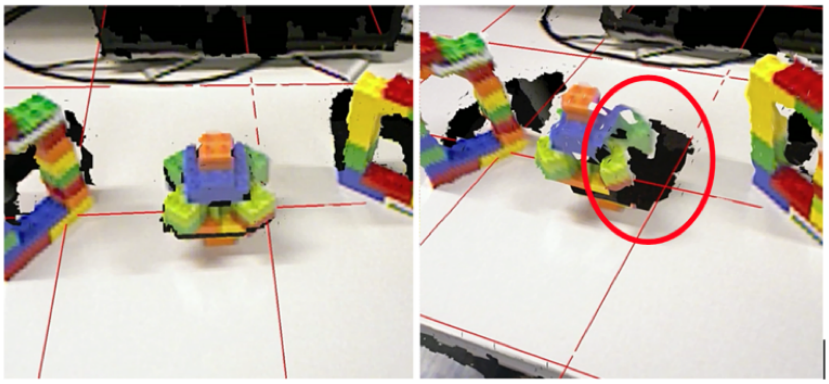


FIGURE 2.12: When the viewpoint (right) deviated significantly from the capture pose of the depth sensor (left), the 3D scene would be largely incomplete and gaps in the dataset became visible (the black parts in the right image) [112]

With the RemoteFusion system, Adcock et al. [1] presented one of the first remote guidance systems that supported an independent 3D viewpoint for the remote expert. On the local worker's side, they used two depth sensors to reconstruct the worker's physical work environment: one was responsible for showing a top-down view of the workspace, and the other one could move around the scene to build up additional details. In addition, a projector was set on top of the work environment to project the expert guidance cues directly on the surface of the local workspace (Figure 2.13). In this case, the local worker did not need to take his/her eyes off the current task to view the remote

guidance. On the remote expert's side, a multi-touch screen was used to allow them to control his/her viewpoint by simply rotating the scene with two-finger touch and zooming in with a pinch gesture. In order to provide guidance, the remote expert could draw on the local worker's reconstructed scene by using a single finger.

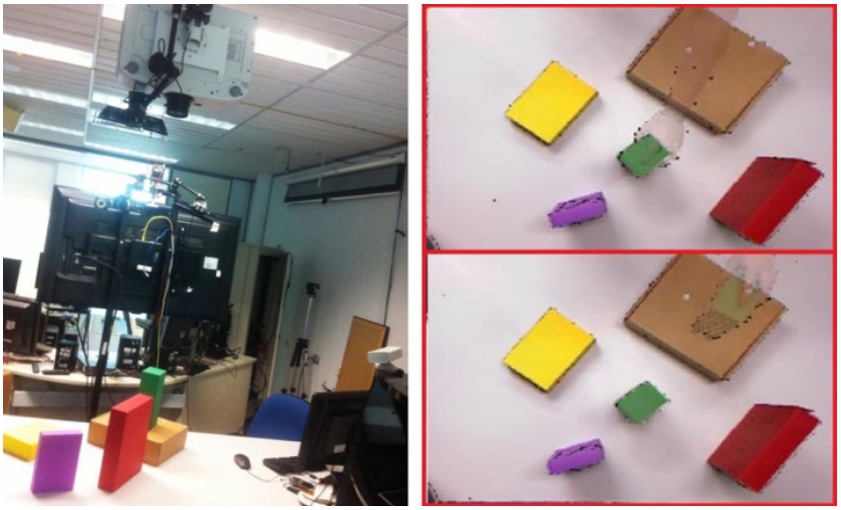


FIGURE 2.13: The worker space of RemoteFusion (left) and dynamic remote guidance projected onto the surface of the local workspace (right) [1]

To provide an independent viewpoint control for remote expert, Sodhi et al. [107] presented a remote collaboration system called BeThere, which allowed remote experts to move freely in a large work environment to perform a variety of virtual interactions. As shown in Figure 2.14, the system was composed of a smartphone, a front-facing Kinect to capture the workspace, a side facing short-range depth sensor to catch spatial gesture input, and a touch strip for more input options.

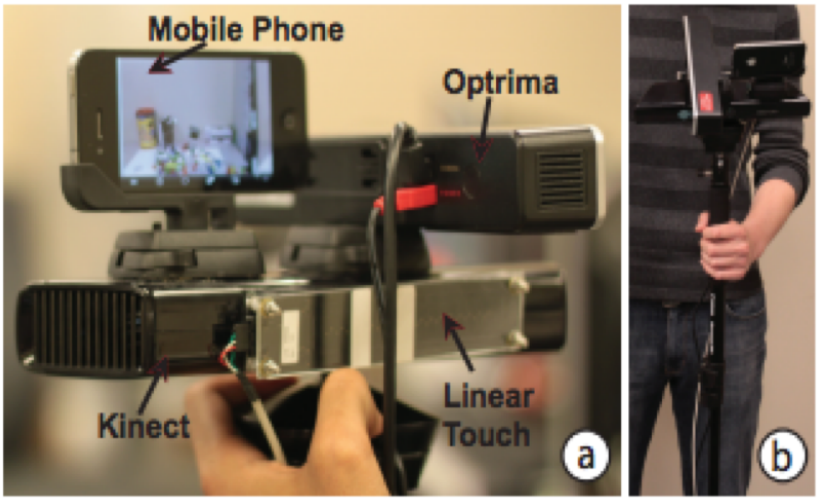


FIGURE 2.14: BeThere remote collaboration system setup [107]

The local worker's physical workspace was captured and reconstructed into a 3D virtual

scene, which was then shared with the remote expert. Hand gestures of both the worker and the expert could be captured by the side-facing depth sensor and then integrated into the reconstructed 3D virtual scene (Figure 2.15). The whole system was completely self-contained and handheld, so the users could interact with the environment from any direction they wanted, offering fully independent viewpoint control for both the worker and the expert. However, one disadvantage of this system was that hand gestures were captured by the side facing depth sensor, so the users might get confused about the spatial relationship between their hands and the objects in the scene and find it difficult to point to the object they wanted. Furthermore, the system relied on the point cloud generated from the depth sensor of the local user, so when the viewpoint of the remote expert was far away from the local worker, he/she could see large black gaps around the virtual objects from missing points in the dataset.

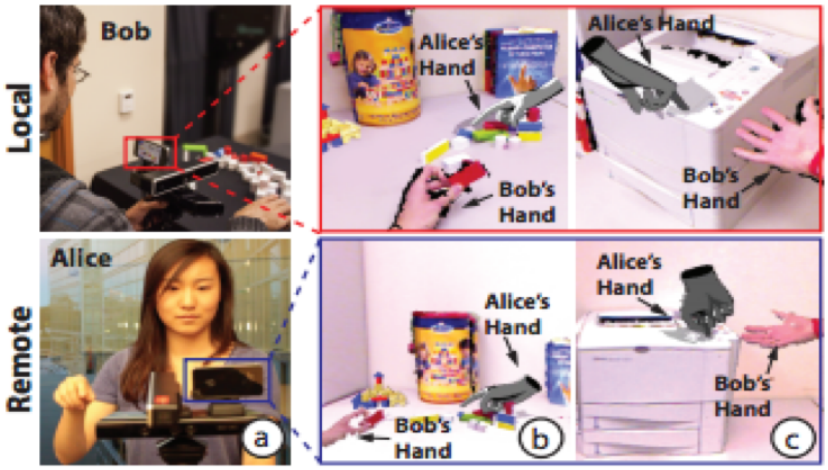


FIGURE 2.15: Remote collaboration by using BeThere system [107]

Besides capturing the local scene in 3D, researchers had also introduced AR and VR methods that enabled remote experts to use 3D virtual annotations for reference. Chastine et al. [17] presented a straightforward method that allowed the remote user to use a 3D virtual arrow in guiding tasks. However, it was time-consuming to manipulate the arrow in 3D space. Bottecchia et al. [10] devised a catalog of pre-defined 3D virtual animations for the expert to choose according to the stage of the task progress; however, the amount of pre-defined animations might not be flexible enough to satisfy the needs of all the situations during the tasks. Other researchers also tried to implement natural visual cues, such as God-like hand gestures [109] and 3D helping hands [112]. However, it could still be challenging to provide precise guidance cues for manipulating the physical targets in the local workspace.

To address the above issue, Oda et al. [89] [26] allowed the expert to create and manipulate virtual replicas of physical objects in the local workspace, and display this manipulation as AR content to indicate actions. They introduced two interface designs. The first one supported pairs of prominent landmarks on the targets for the worker to align them (Figure 2.16a), while the second one supported 6DOF manipulation of the virtual replica for direct instruction (Figure 2.16b). Compared to 2D based annotation, user study results showed that their methods enabled a highly trained expert to guide a local worker completing tasks much faster, especially for the first approach using pairs of prominent landmarks.

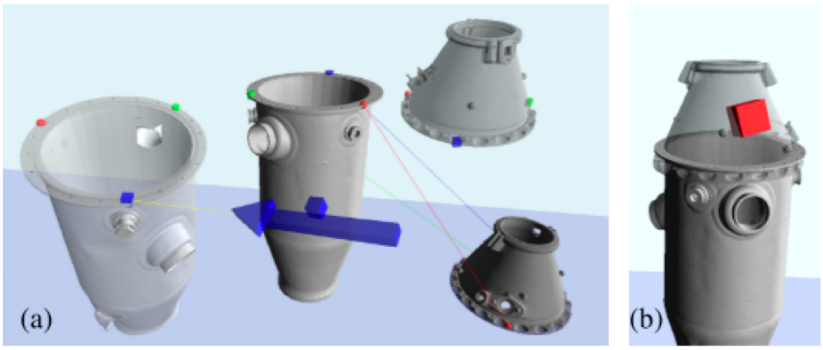


FIGURE 2.16: Virtual replicas for remote assistance [89] [26]

Based on these previous research, we saw that although these remote collaboration systems enabled remote experts to help workers over a distance, most of them required both the expert and the worker to collaborate in a small workspace. This might not meet the requirements of some collaborative tasks in a practical work environment. For example, for the maintenance of sophisticated machines in a semi-automated factory, machines and tools may be placed around a large room that requires the worker to walk around to interact with the target objects. In this case, it may be better to use a dynamic remote collaboration system that could allow the worker to capture their whole workspace.

2.2 Scene Capture for Room-scale Remote Collaboration

Most previous remote collaboration systems reviewed in section 2.1 shared video of the local worker's workspace with the remote expert. In many cases, this video was captured from a head-mounted camera worn by the local worker, which limited the remote expert's point of view to be exactly the same as the local worker. To overcome this limitation, some researchers had begun to explore how 360° video [58] [106] and

panorama imagery [105] [9] could be shared to allow the remote user to have an independent view into the remote space. Other researchers focused on how depth sensors could be used to create a 3D reconstruction of the entire local worker's workspace, enabling the remote user to view the space in 3D and have a completely independent perspective [111].

Previous research showed that 360° video could support better immersive experiences for users by enabling them to share their surrounding environment and current activities. The global market of 360° cameras was projected to grow around 35% per year between 2016 and 2020 [41], so such cameras have become more common and cheaper. With the support of popular commodity HMDs, 360° video-sharing showed significant potential advantages for remote collaboration in immersive VR environments. However, there were some challenges that researchers needed to overcome before bringing it to the commercial market.

The first issue that needed to be addressed for 360° video sharing is shown in Figure 2.17. Full 360° videos contained much more information than the viewer could see since HMDs only supported a limited field of view [28]. Therefore, streaming 360° videos in full resolution was considered a waste of resources, such as network bandwidth, processing power, and storage space, which might degrade the user experience. One solution streamed the view of the current FOV (enlarged with heuristic factors) in full resolution and the entire 360° video at a reduced resolution [82] [45] [63]. While the viewer rotated his/her view outside the streamed current FOV, the system would display the reduced-resolution video to the viewer as one alternative approach until the next full-resolution FOV arrived. The other solution was to predict the viewer's FOV for the next frame in advance and only transfer the region of 360° video that has a high probability of being viewed [28] [81].

FIGURE 2.17: Issues of 360° videos sharing: (a) Full 360° videos contain rich visual information; (b) a viewer can only see a small part of the video at any one moment [28]

The other challenge for 360° video sharing was how to support continuously focusing

and re-focusing on the intended targets. Full 360° videos contained rich information for the viewer to discover. However, while watching one specific fast-moving target, the viewer might quickly lose track (the issue of continuously focusing) and need to search for the target again in the 360° video (issue of re-focusing). Researchers had introduced two possible focus assistance techniques. One followed the moving target by automatically changing the current FOV [79] [44] [7], which took the viewer directly to the intended target (Figure 2.18 A). The other used an augmented visual indicator to guide the viewer to the target region [79] (Figure 2.18 B). Both of these methods could improve the ease of focus while watching 360° videos, but the advantages of each approach depended on not only individual differences but also the video watching goal.

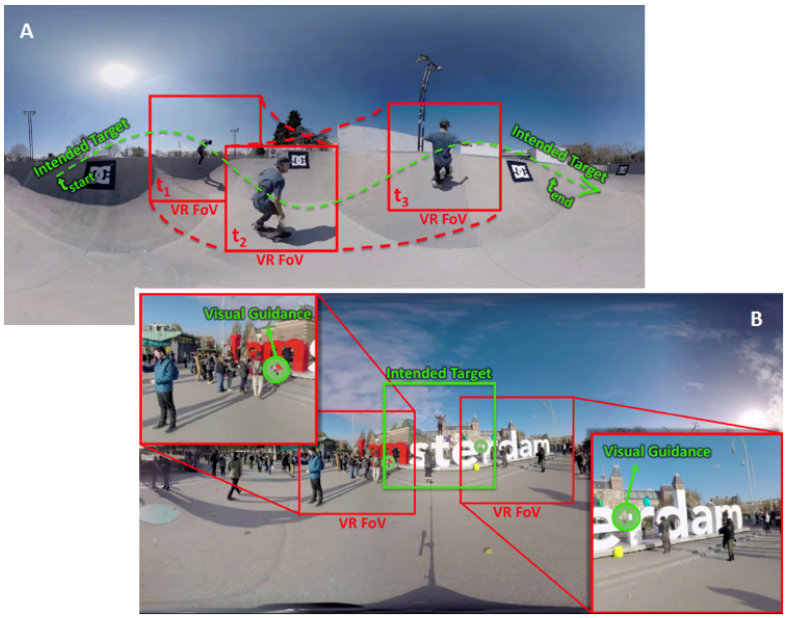


FIGURE 2.18: A: Auto Pilot: auto-change the current FOV following the moving target; B Visual guidance: use augmented cues to indicate the target [79]

Based on 360° videos sharing techniques, Gun et al. [75] [74] presented their MR remote collaboration system called SharedSphere (Figure 2.19). On the local side, the user wore one 360° panorama camera to capture the surrounding environment and shared the view to an expert in a remote location. In order to view the shared live panorama scene, the remote expert wore a VR HMD. The hand gestures of the remote expert were captured and used as non-verbal communication cues to guide the local user. The user feedback from their study indicated that the live 360° video created an immersive environment for rich information sharing between the local user and remote expert during remote communication. However, since the video was shared as a 2D panorama image, the remote expert could only rotate his/her view away from the local user's original viewpoint but without the ability to move to another viewpoint. In this case,

view independence had been limited. In order to support a truly independent view for both local and remote users, researchers started to capture the large-scale local scene in the form of 3D contents by using depth sensors.

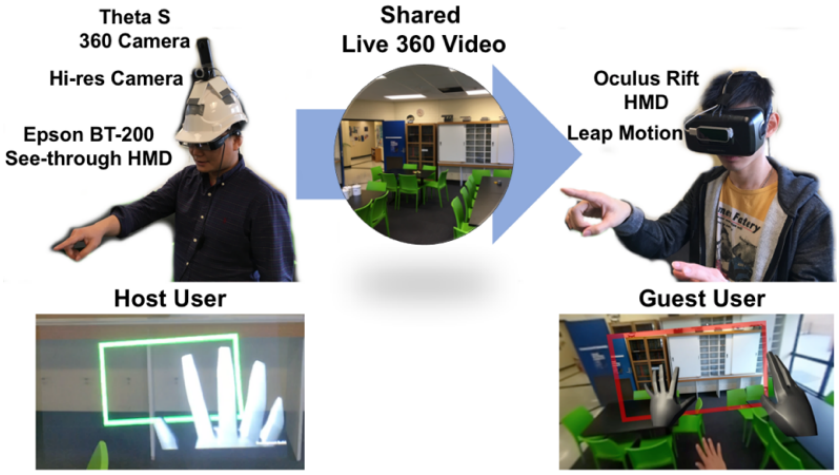


FIGURE 2.19: SharedSphere system overview [75]

Using depth sensors to create real-time 3D scene reconstruction has been studied by researchers for many years. Izadi et al. [53] presented a novel interactive reconstruction system called KinectFusion, which allowed a real-time volumetric dense reconstruction of a desk-sized scene at sub-centimeter resolution. A user could reconstruct the 3D model of the physical scene within seconds (Figure 2.20 B, in which the wireframe frustum shows the current tracked 3D pose of the Kinect) by holding a standard Kinect camera (Figure 2.20 A) and moving within any indoor space. The system could continuously track the six degree-of-freedom (DOF) pose of the camera and fuse new viewpoints of the scene into a global surface-based representation.

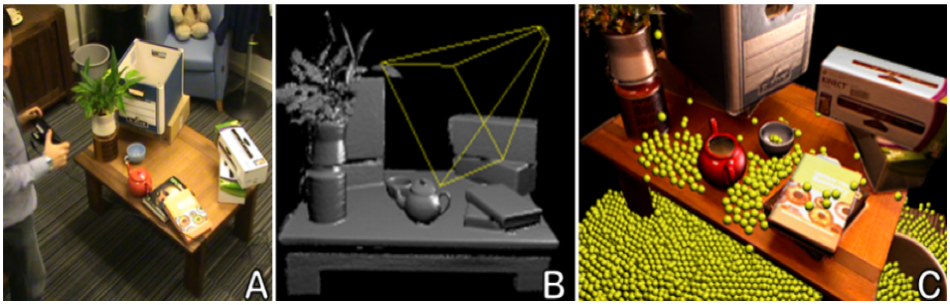


FIGURE 2.20: KinectFusion enables real-time detailed 3D reconstructions of indoor scenes using only the depth data from a standard Kinect camera [53]

They indicated the key uses of KinectFusion as a low-cost handheld scanner to support novel interactive methods for segmenting physical objects of interest from the reconstructed scene (Figure 2.20 C). KinectFusion enabled scene reconstructions at an

unprecedented quality at real-time speeds, but still faced some limitations. For example, it only supported tracking in a fixed small area, used geometric information as the single parameter to estimate camera pose, and did not explicitly support incorporating loop closure. These three limitations restricted the applicability of KinectFusion to large-scale SLAM problems.

A number of derived works have been published recently after the advent of the KinectFusion system. Bylow et al. [14] directly tracked the camera pose by representing the geometry with a signed distance function, which was more accurate and robust than the iterated closet point algorithm (ICP) used by KinectFusion. Roth and Vona [99] extended the operational range of KinectFusion by automatically translating and rotating the volumetric models through space as the camera moved. Zeng et al. [124] presented a memory-efficient implementation of KinectFusion based on an octree representation that allowed mapping of areas up to 8x8x8m in size.

There are still many challenges in the 3D reconstruction of dynamic objects, such as capturing the large-scale scene, increasing the capture resolution, and reducing the holes caused by dynamic occlusions. In order to overcome these issues, Chabra et al. [16] showed an approach by optimizing the placements of the depth sensors and using event-specific optimization weights to achieve better capture results. They indicated that their algorithm not only enhanced the total coverage of the reconstruction but also filled the voids when compared to manual sensor placements (Figure 2.21).

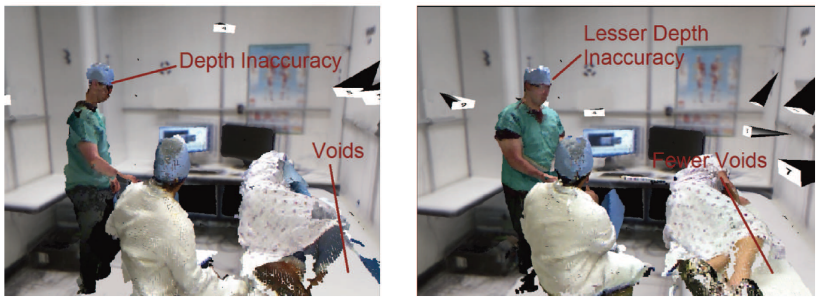


FIGURE 2.21: The reconstructed point clouds generated from captures with manually-designed and with optimized camera placements [16]

Dou et al. [24] presented a 3D capture system to build a complete and accurate 3D model for dynamic objects by fusing the 3D data captured from depth sensor into a 3D model, and then track the model to match the following captures. However, their system faced some limitations, such as the segmentation issue and body topology change issue. To address these issues, Dou et al. [25] introduced another real-time 3D reconstruction system called Fusion4D, which allowed for incremental nonrigid reconstruction from

noisy input based on multiple RGB-D sensors. As shown in Figure 2.22, this system enabled the robust capturing of dynamic objects to many complex topology changes. Based on Dou et al.'s work, Orts-Escolano et al. [93] presented their Holoportation system enabled high-quality, real-time 3D reconstruction and sharing of an entire space, including people, furniture and objects, using eight depth sensors in total. They claimed that this system required high-end hardware, such as GPU-powered PCs and 10 Gigabit Ethernet connection, to support low end-to-end communication and remote rendering latency for MR telepresence.

FIGURE 2.22: Fusion4D is robust to many complex topology [25]

Another technology, called InfiniTAM [57], also supported dynamic 3D scene capturing of large-scale workspace, which allowed for the rapid capture and reconstruction of 3D space using handheld devices attached with depth sensor (Figure 2.23). The latest version of InfiniTAM supported loop closure detection [56] and surfel-based reconstruction [96]. However, this has not yet been applied to remote collaboration since it also required significant computing capability and advanced hardware setup.

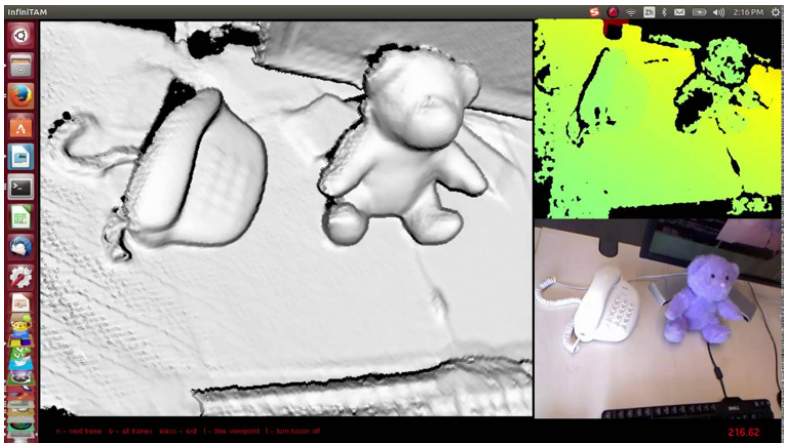


FIGURE 2.23: InfiniTAM 3D scene reconstruction [57]

Reconstructing a large workspace as a 3D virtual scene with real-time updates of environment changes is still a challenge for researchers using current hardware and software. In my research, I attempted to overcome this limitation by capturing the local scene as a static 3D model and providing a high-resolution 2D video to show the local workspace's real-time changes.

2.3 Mutual Awareness Cues for Tele-presence Systems

Shared Collaborative Virtual Environments (CVEs) were used to simulate the face-to-face physical environment to reduce the cognitive load between users attending from different locations [121]. However, current remote collaboration systems were limited in how they could enhance perception and cognition. To solve this issue, researchers tried to find approaches that could support two mutual-awareness mechanisms [27]: (1) mapping the position of the user's annotations or notes, and (2) mapping the partner's position. Most previous remote collaboration research focused on studying the first mechanism by supporting multiple kinds of communication cues, such as a pointer, annotations, and gestures. However, there was little work on solving the second issue of distributed cognition between users, which might lead to incorrect spatial faithfulness and decrease remote collaboration efficiency.

Correct spatial faithfulness could increase the degree of telepresence for simulating face-to-face collaboration. According to Nguyen and Canny's study [88], there were three levels of spatial faithfulness:

- Mutual spatial faithfulness, which enabled users to receive information from their partners;
- Partial spatial faithfulness, which enabled users to map the perceived action with the partner's actual action in the correct way;
- Full spatial faithfulness, which enabled world-stabilized action mapping between the users and captured objects.

As shown in Figure 2.24 a, face-to-face collaboration supported full spatial faithfulness since users could directly communicate with each other based on multiple perceptual cues, such as the sense of vision, audition, touch and smell [121]. On the other hand, the capture and display devices of remote collaboration systems were usually placed in inconsistent positions, which might lead to an incorrect impression of the remote actions, and only partial spatial faithfulness was supported. For instance, in Figure 2.24 b, three

users worked together remotely. Based on the individual camera and display screen setup, each user felt the other two partners were looking at him or her; however, in fact, they were all looking at the target box. In this case, it required additional cognitive processes for the users to correctly map their partners' actions into a consistently shared world space.

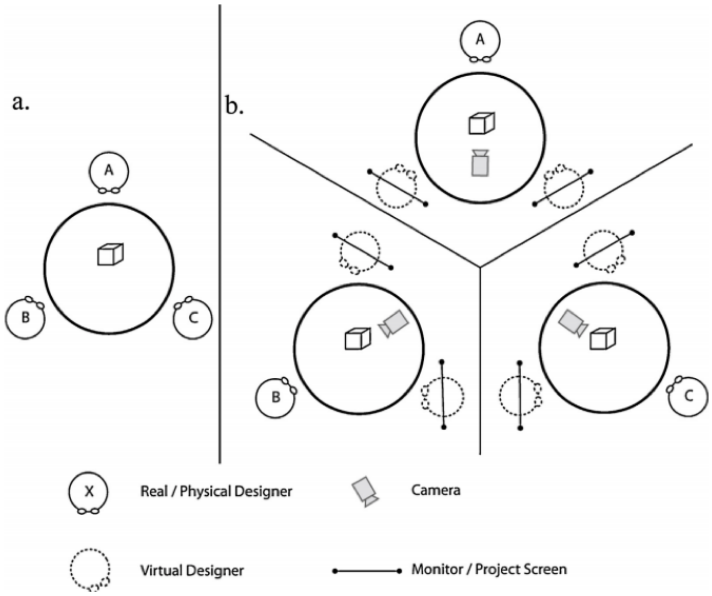


FIGURE 2.24: Collaboration system setups: (a) face-to-face collaboration; (b) remote collaboration [121]

Distributed cognition theory [98] [100] indicated that human cognition depended on not only the individuals' brains but also the interrelation with other humans or objects in the working environment. To support distributed cognition, the ClearBoard system [52] [51] allowed real-time remote eye contact through video by using a half mirror polarizing film projection screen. Similarly, the Blue-C system [43] introduced 3D projection technology into a CAVE-like environment to provide a spatially immersive experience. As a further improvement of previous research, the DigiTable system [21] brought distributed users together by using shadows. As shown in Figure 2.25, the left user could detect the right user's hand with the shadow projected on the table surface (left bottom image), and vice versa. In this case, both users had the ability to maintain awareness of their partner's actions.

Telepresence is the feeling of being present in a remote environment, and telepresence systems can be used to create a place for users immersing themselves together while doing the same tasks [114]. Thalmann et al. [114] identified three kinds of telepresence technologies: (1) Realistic 3D telepresence systems brought users to a reconstructed remote place; (2) Networked 3D VR and AR enabled users from different locations to

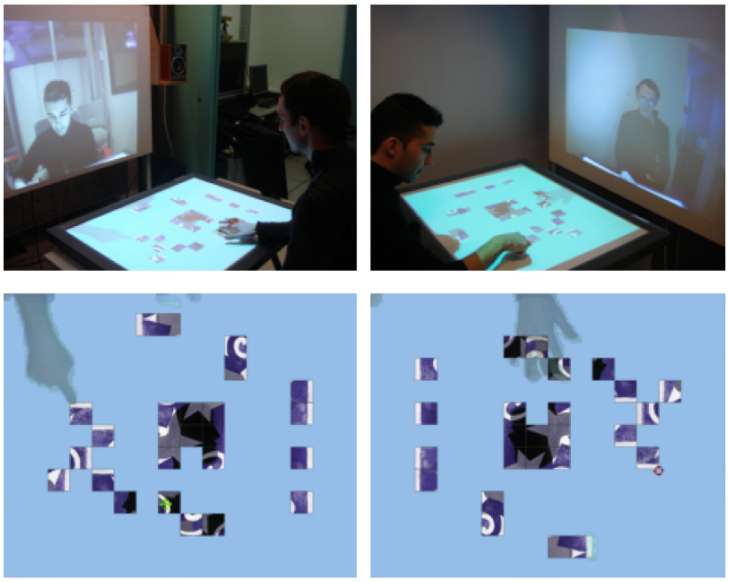


FIGURE 2.25: DigiTable system

work together in a shared virtual/augmented space; (3) Telepresence robots enabled users to operate robots in a real remote environment. The second approach showed the possibility of supporting users to perceive a partner's actions remotely, which could be considered a solution for solving distributed cognition.

A users' body actions and environmental changes could be captured and mapped into a networked collaborative virtual environment using avatars and reconstructed 3D models. Furthermore, this environment could be shared with all distant users involved in the task. In this case, one user could directly see, hear, and feel the other partners, which was similar to the work process during face-to-face collaborative tasks. To create avatars based on the human body, Lee et al. [76] introduced one approach that automatically mapped photos taken from real people as textures onto virtual human models (Figure 2.26).

In interactive virtual environments, real-time avatar tracking provides users with an intuitive and natural way to communicate with other players. Depth cameras are an inexpensive alternative compared to expensive and cumbersome marker-based motion capture systems used for body posture or hand gesture recognition. To create a realistic and robust avatar system in VR, skeleton tracking is one of the most critical steps that need to be considered, as it is the primary data source for the avatar rig.

The Microsoft Kinect v2 [1] could track the skeletons of multiple users in real-time but had problems with occlusion. In addition, the Kinect v2 was not able to recognize

[1] https://developer.microsoft.com/en-us/windows/kinect

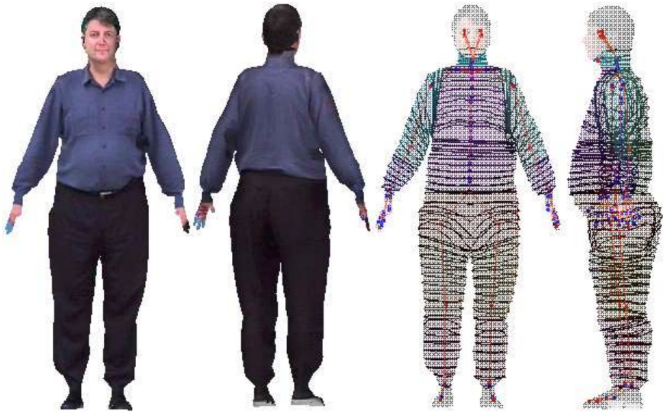


FIGURE 2.26: Full body textures mapping with detailed individualized faces

which way a person was facing (either forwards or backwards), which decreased the pose and motion accuracy accordingly while the person kept moving or turning. A multiple-Kinect solution could address this issue by integrating the data from each Kinect to optimize the accuracy of the skeleton created. The main solutions for multiple Kinects to date could be divided into two groups: a one-to-many mode and a network mode. For the first approach, Mokhov et al. [84] used three Kinect sensors in one single application to find and track a user's skeleton data. The network mode used one PC as the server to receive data for client Kinects through the network. Muller et al. [85] placed six Kinect v2 sensors along a corridor with three Kinects on each side to capture a users' motion with 3D reconstructed models. Kitsikidis et al. [66] fused skeleton data from multiple Kinect v1 devices for dance analysis. They used the ICP algorithm [19] to estimate the rigid transformation for calibrating multiple cameras. This method could be used for data fusion and was good enough for motion pattern recognition. However, they did not show the accuracy of the fused data, as the tracking state of some joints might be different from each Kinect.

Real-time full body capturing was still a challenge with current hardware setups and network conditions. In my remote collaboration system design, I tried to use a virtual view frustum and a head avatar to present each user's head position and orientation in the shared virtual environment in real-time (Section 4.1). By doing this, users could have awareness of their remote partners' current focus, which might solve the issue of distributed cognition.

2.4 Conclusion

In this chapter, I have reviewed some previous examples in the field of remote collaboration system design, and discussed the advantages and limitations of each of them. Based on this literature review, we could find some general answers to the key questions of remote collaboration system design that I have discussed in Section 1.1.

A first-person view of the local workspace was beneficial for the remote expert to understand the local worker's current focus no matter if it was in the form of a 2D video or a 3D scene. However, when the local worker changed his/her view too frequently, it might interrupt the guidance process. Therefore, independent viewpoint control was an important feature for remote collaboration system design, especially for large-scale workspaces. To deal with this issue, 3D capture of the local scene shared in the VR world with multiple users seemed to be a solution. It was also useful for enhancing spatial awareness because it enabled the remote expert to experience the distance between objects from the local side during collaborative tasks. However, integrating multiple real-time 3D reconstructions into a remote collaboration system was still a challenge for researchers.

In the following chapters, I presented my alternative solutions for room-scale remote collaboration that combined static 3D reconstruction and different types of live feedback to support collaborative tasks. In my solution, the static 3D reconstruction supported remote experts with the straightforward spatial awareness of the local physical environment, and the live feedback enabled the experts to monitor task processes in real-time, in which case the system could ideally solve the issue of independent viewpoint control. Finally, I also used a variety of VR visual representations to provide guidance cues and improve mutual awareness.

Chapter 3

Spatial Awareness for Mixed Reality Remote Collaboration

In this chapter, I report on a prototype system for MR remote collaboration. Our remote collaboration system can support hand gestures used by a remote expert, or other visual aids to be shared with a remote worker to assist him/her in performing manual tasks. Traditional remote collaboration systems often use video cameras on each side to combine the views of both workspaces together and display the result on desktop monitors or handheld devices. In this case, the remote expert's virtual guidance cues can be directly overlaid on the video of the worker's workspace to help the local workers finish their physical tasks. However, recent research showed that it could be difficult for the remote expert to understand the spatial relationships between the objects in the local worker's workspace while using a 2D desktop display [50] [112]. Furthermore, 2D remote guiding systems may not support the sharing of complex hand gestures to enhance collaboration efficiency.

In order to address these problems, we have developed a prototype system that used VR HMDs and depth sensors to capture the 3D surrounding environment of the local worker and map the current point cloud data into the VR world, helping the remote expert better understand spatial relationships during a physical task. The hand gestures of the remote expert were also detected and shown in the 3D space to improve the communication. Using the prototype interface, we conducted a user study to evaluate the benefits of our system and discuss the feedback from study participants in detail.

3.1 Oriented View MR Remote Collaboration

We present an MR system for remote collaboration using VR headsets with an external depth camera attached. Our system wirelessly shared the 3D point cloud data of a local workers' workspace with a remote expert, and shared the remote expert's hand gestures back to the local worker, enabling the remote expert to assist the worker to perform manual tasks. Displaying the point cloud video in a conventional way, such as a front view in a VR headset, may not provide the expert with sufficient understanding of the spatial relationship between their hands and the remote surroundings. In contrast, our MR system shared with the remote expert, not only the full 3D captured environment data but also real-time orientation info of the worker's viewpoint. To accomplish this, our system combined single-frame point cloud capture data and free-hand tracking, as described in the next sections.

3.1.1 Single-frame Point Cloud Capture

In order to gather depth information for the extracted features, we first combined a SoftKinetic DS325 depth sensor (see Figure 3.1 d) with a VR headset (Oculus Rift DK2). This sensor provided short-range depth detection at a distance from 0.15 meter to 1 meter. The OpenNI2 library [1] was used to grab depth information from this sensor. However, the original driver of the DS325 sensor did not support the depth and RGB image alignment; therefore, a point from the color frame could not be correctly projected into the depth map. From Figure 3.1 a and Figure 3.1 b, we can see that depth map has a wide view range than the color map.

In order to achieve point-to-point projection from the depth map to the RGB image, we used the camera intrinsic matrix K, and extrinsic parameters R and T, defined as:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

[1] https://structure.io/openni

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

Where:

- f_x, f_y are the focal lengths in pixel-related units along the x and y axes respectively;
- c_x, c_y are the principal point of the camera along the x and y axes;
- R is the rotation matrix between the color and depth cameras;
- T is the translation vector between the color and depth cameras expressed in meters;

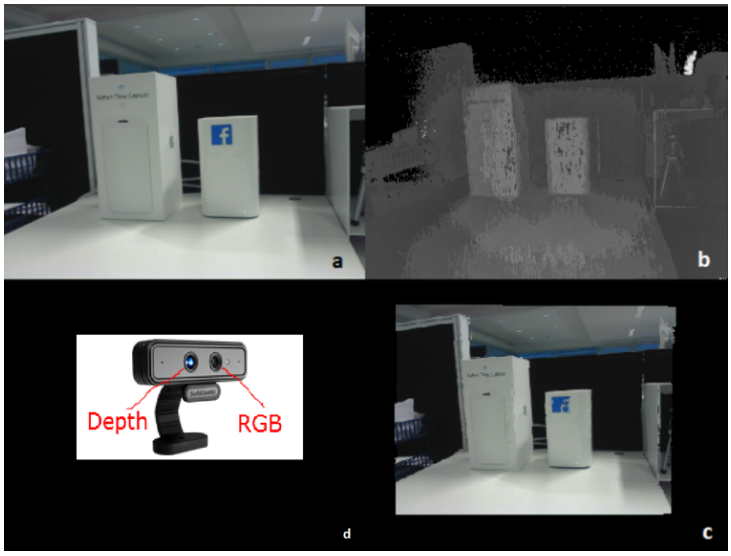


FIGURE 3.1: Depth and RGB image alignment: (a) RGB frame; (b) Depth frame; (c) Mapped RGB frame; (d) the DS325 depth sensor. Showing that (b) has a wider field of view than (a)

The DS325 driver provided functions for grabbing these matrices, so we did not need to calibrate the camera by ourselves. Assuming p_{depth} was a pixel in the depth map,

the 3D point P_{depth} in the space corresponding to the p_{depth} could be calculated by back-projecting p_{depth} in the depth sensor's coordinate system, as shown by the equation 3.1:

$$P_{depth} = inv(K_{depth}) * p_{depth} \tag{3.1}$$

Then, P_{depth} could be transformed to the RGB camera's coordinate system through the relative transformation R and T (equation 3.2).

$$P_{RGB} = R * P_{depth} + T \tag{3.2}$$

Finally, we projected the 3D point P_{RGB} onto the RGB camera image to obtain the corresponding 2D coordinate.

$$p_{RGB} = K_{RGB} * P_{RGB} \tag{3.3}$$

Figure 3.1 c shows the calibration results, showing the two datasets were aligned together. The frame rate of this calibration system was up to 30 fps which satisfied the requirement of our real-time 3D capturing algorithm.

In this case, for each point on the RGB image, we could find its corresponding depth information form the depth map and receive a point coordinate as $p\{x_p, y_p, z_d\}$, in which x_p and y_p were the pixel coordinates and z_d was the depth in meters. Based on equation 3.4, we could transform one pixel's 3D coordinate in meters.

$$P_{RGB} = inv(K_{RGB}) * p_{RGB} \tag{3.4}$$

Then, we could eventually map the color and depth maps as one point cloud in the VR environment by projecting each pixel on the color map to the 3D space. Figure 3.2 shows one example of single frame point cloud capture.

3.1.2 Free-Hand Tracking

In addition to capturing a point cloud of the local user's workplace, our system performed free-hand tracking (no marker or gloves required). Skin color has been proven to be a robust cue for the human face and hand detection [119]. Color detection allows for

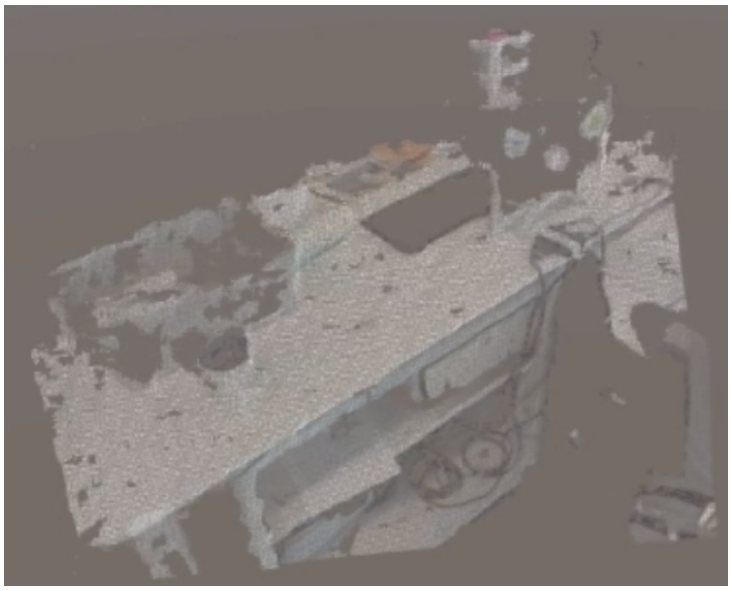


FIGURE 3.2: Simple single frame point cloud capture

fast processing without the need for high-powered computing or graphics processing. The human skin has a characteristic color that can be recognized by computers after image processing with related color spaces [119].

Skin color detection aims to separate all of the pixels of an image into two parts: skin and non-skin pixels. There are three possible and widely used approaches for segmenting the skin color area from an image: using an explicitly defined skin region [68] [2], using a nonparametric skin distribution model [11] [42], and using a parametric skin distribution model [92] [108]. The fastest and most accurate way to segment skin color area from an image is to build a skin classifier based on a group of training images through a set of rules [11]. However, the performance of this method directly depends on the training images collected. If the set of the training data is too small, the performance may be even worse than the method using explicitly defined boundaries.

We aimed to use tracking methods based on an explicitly defined skin region since they could provide real-time hand detection. These methods were carried out in different color spaces based on explicitly defined boundaries, such as the RGB, HSV, and YCrCb color spaces. In our research, we combined the Otsu threshold algorithm [94] and the YCrCb color space together to segment the hand region. In image processing, the Otsu algorithm was used to threshold the input image into two classes of pixels (the skin color and non-skin color) and output it as a binary image. The input image was first transferred from the RGB color space to the YCrCb color space. Then the Otsu algorithm was applied to segment the skin color region. From the example shown in Figure 3.3, this approach appeared more robust than the others and was not affected by the light.

By cutting the background off with a depth classifier (e.g., remove the points with a distance to the camera further than 1 meter), we could detect and track the hand region in real-time with 30 fps.

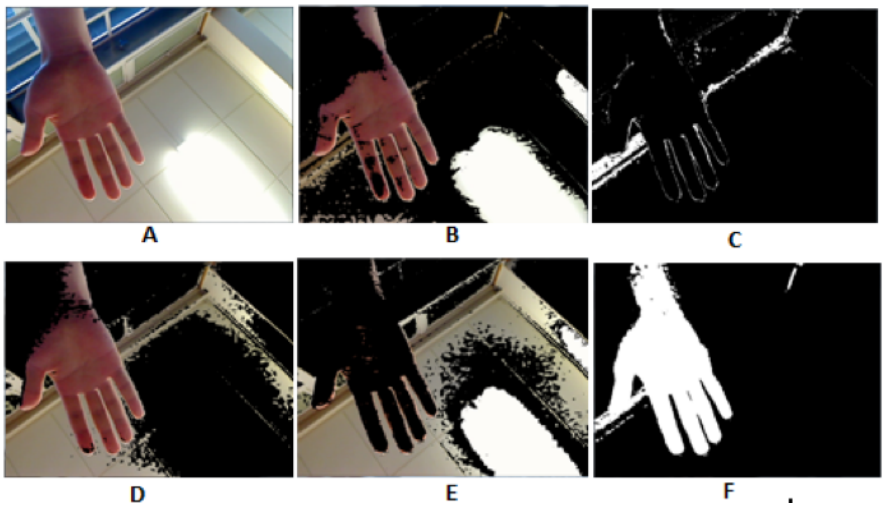


FIGURE 3.3: Hand region detection: Input image (A); RGB colour space (B); Normalized RGB (C); YCrCb colour space (D); HSV colour space (E); YCrCb and Otsu threshold (F)

3.1.3 Prototype System Setup

Our prototype connected two separate workspaces: a local space for a worker to work on a physical task, and a remote space for an expert to provide guidance to the local worker (Figure 3.4).

FIGURE 3.4: System setup: remote expert using hand gesture to guide the local worker

In the worker's space, the user performed a physical task while wearing a VR headset (the Oculus Rift DK2). In order to enable the user to see the surrounding scene, a depth sensor (Soft Kinetic DS325) was attached to the front face of the worker's headset, facing towards the workspace. The system could capture the current scene based on the aligned RGB frame and the depth frame from the sensor, and map the current frame into the VR world coordinate system as a dense point cloud. We rendered this point cloud view in the VR headset for the worker as a video see-through display. In this case, the local worker could check his surrounding environment and interact with the physical objects on the table.

The expert wore the same VR headset as well. The scene was captured and compressed on the worker's side and streamed to the expert's side for him/her to watch and understand the task situation and local environment. Meanwhile, the depth sensor attached to the expert's headset detected the expert's hand region based on its color frame and presented the hands as a point cloud based on its depth frame (see Figure 3.5). The hand point cloud pixels were colored based on their different depth values. In this case, the users could quickly identify which hands are the local worker's hands and which hands are the remote expert's hands.

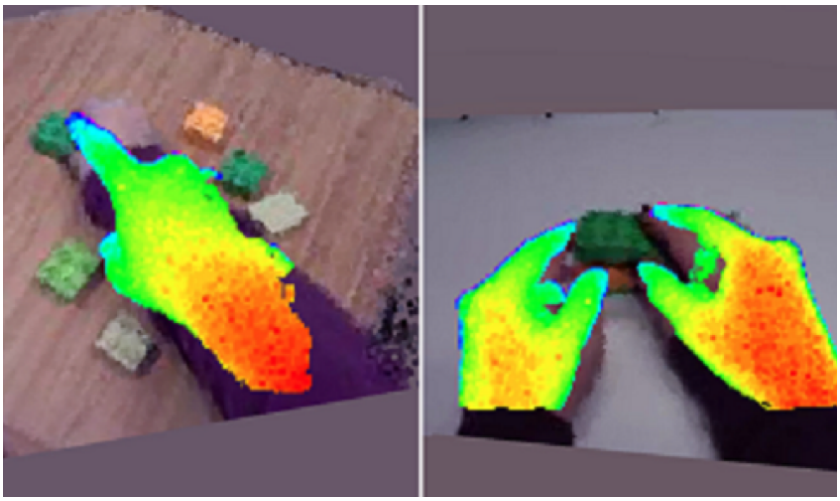


FIGURE 3.5: The remote expert's hands were colored and overlaid on top of the worker's view to guide LEGO assembling. Hand pixels closed to the user were colored red and those furthest away were colored blue.

Both the acquired 3D point cloud of the local worker space and the hand gestures on the remote expert's side were then fused together in real-time, mapped into a single shared virtual coordinate system, and displayed in the VR headsets of both users (Figure 3.4). Furthermore, the orientation data from the built-in gyroscope sensor in the local worker's HMD could be accessed and shared wirelessly from the worker to the remote expert. The remote side application would apply this data directly to the shared point cloud

view, so the point cloud rendered in the VR space was always located identical to where it was in the local physical world. In this case, the remote expert also needed to rotate his/her head with the headset to follow the local user's viewpoint.

Finally, the remote expert could guide the worker to finish the task by using his/her hand gestures such as finger-pointing and hand movements. The user in the worker space could understand what to do by following the hand gestures and speech of the remote expert.

Each side of our system ran on a 64-bit Windows 7 system with an independent desktop PC (Intel Core i5, 8 GB RAM, and NVIDIA GeForce GTX770). The Unity game engine was used for rendering the 3D point cloud scenes, and the AllJoyn framework [2] was used for network data communication between users. Due to the depth sensor's limited resolution, we rendered the point cloud at a resolution of 320 by 240 pixels. Both sides' frame rates could reach 30 fps. The frame rate we used here is related to the system's processing rate, not the display refresh rate of the VR headset.

3.2 User Study

3.2.1 Experiment Setup

In traditional remote video sharing, 2D output devices such as a monitor or tablet were usually used, showing the shared local worker's view to the remote expert without any additional orientation cues. In contrast, our system shared a 3D view in an HMD where both users could have their own independent viewpoint control. Furthermore, the captured single-frame local point cloud was rendered identical to where it was in the local physical world with extra orientation information enabled. We conducted an initial study to investigate the usability and social presence of the way that our system presented 3D visual data to the remote expert. We hypothesis that

- Rendering the local single-frame point cloud scene as an oriented view could increase the remote experts' spatial awareness in terms of social presence in remote collaboration.

We used a within-subjects user study design. Participants were recruited for the role of remote experts, and the experimenter took the role of the local worker. Ten people took part in the study, four men and six women, aged 24 to 34 years old. Most of them had previous experience using VR interfaces and had tried 2D conferencing systems such as

[2] http://allseenalliance.org/framework

Skype on a monthly basis, but with minimal 3D remote collaboration experience. Each participant was asked to guide the local worker to assemble two sets of LEGO blocks (Figure 3.6) into an integrated toy using a front view and oriented view respectively. The two interfaces (Figure 3.7 and Figure 3.8) were provided to the participants in a random order to exclude potential learning effects.

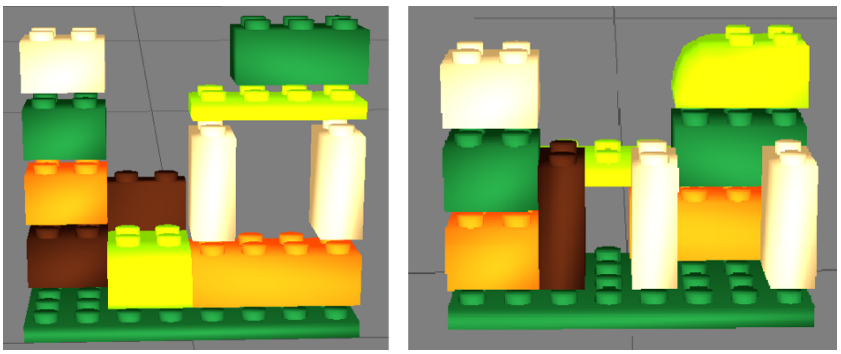


FIGURE 3.6: Two sets of LEGO blocks

There were two different viewing mode conditions [3] that we compared: (1) Front View, and (2) Oriented View. As shown in Figure 3.7, for the Front View mode, the remote expert could see the worker's view without head rotation, so the experience was similar to watching a TV hanging in front of his or her view. For the Oriented View mode (Figure 3.8), if the remote expert wanted to see the worker's view, he/she needed to rotate his/her head towards the same direction of the worker's head, which helped the remote expert to understand better the spatial layout of the worker's surrounding environment.

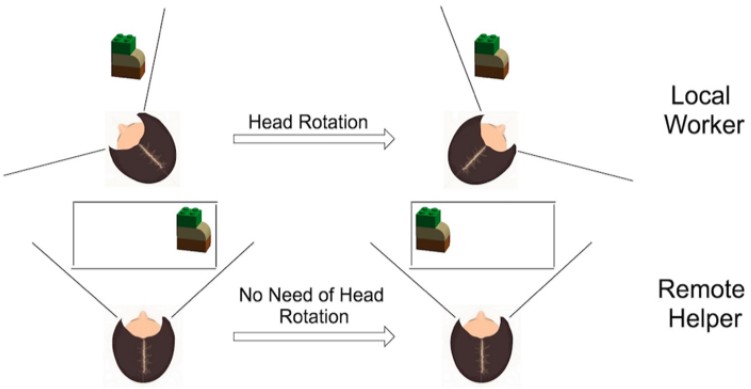


FIGURE 3.7: The Front View mode

Since the expert's vision was blocked by the VR headset from the physical world, we showed a 3D assembly model of the LEGO toy in the remote expert's virtual view during

[3] Video link of two viewing modes: https://www.youtube.com/watch?v=SIv68Hx9Pys&t=8s

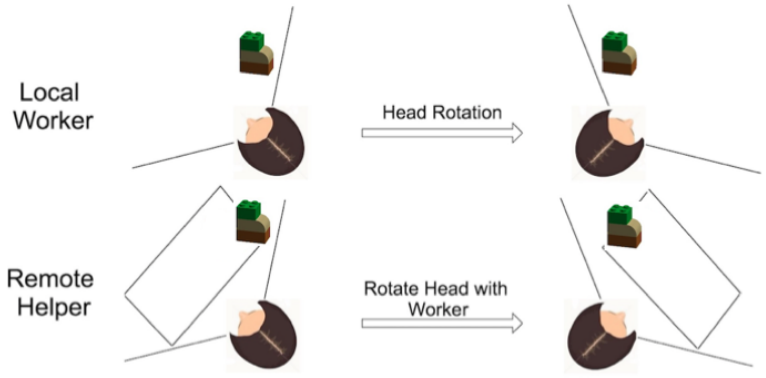


FIGURE 3.8: The Oriented View mode

the task (Figure 3.4, right bottom). The expert could use the mouse wheel to rotate the model to check the assembly details. The session was considered complete when the toys were correctly assembled for both viewing mode conditions. After the participants completed each condition, they were asked to provide feedback by answering the usability and social presence related interview questionnaire using a five-point Likert scale (1 to 5 with 1 indicating strongly disagree while 5 indicating strongly agree). In addition, we also asked participants to provide comments in response to questions in the post-experiment questionnaire.

3.2.2 Experiment Result and Discussion

The results for usability custom rating items are shown in Figure 3.9. Of the 10 participants recruited to the study, the Oriented View mode elicited an increase in physical stress in seven participants compared to the Front View mode, whereas two participants saw no difference and one participant felt more physical stress with the Oriented View mode. A Wilcoxon signed-rank test determined that there was a statistically significant median increase in physical stress when subjects using Oriented View mode (median = 3, interquartile range = 2.250) compared to using Front View mode (median = 2, interquartile range = 1.000), $z = 2.124$, $p = 0.034$. No significant difference was found for the rest usability ranking questions (Like to use: $z = -0.966$, $p = 0.334$; Easy to use: $z = 0.175$, $p = 0.861$; Well integrated: $z = 0.176$, $p = 0.860$; Helpful: $z = 0.333$, $p = 0.739$; Mental stress: $z = 0.000$, $p = 1.000$).

We used three sub-factors from the social presence questionnaire (SoPQ) [48], which included Co-presence (CP), Perceived Message Understanding (PMU), and Perceived Behavioral Interdependence (PBI). Each of these sub-factors consisted of six closely interrelated questions scaled from 1 (strongly disagree) to 5 (strongly agree), representing

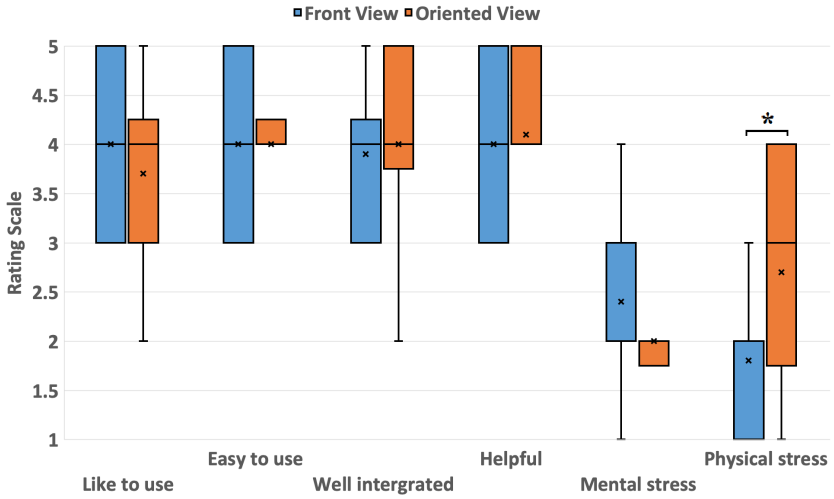


FIGURE 3.9: Results of the usability rating scale (∗: statistically significant difference)

a group of Likert scale measurements. To analyze Likert scale data, we first calculated a composite score from the six questions for each sub-factor [110], then a paired-samples t-test was used to determine whether there was a statistically significant mean difference between the two view modes. As shown in Figure 3.10, we found no significant difference in term of overall social presence (t(9) = 0.532, p = 0.608). We also analysed each sub-factors and found a significant difference in CP (Front View mode: mean = 3.975, standard deviation = 0.533; Oriented View mode: mean = 4.275, standard deviation = 0.399; t(9) = 2.449, p = 0.037), but not in PMU (t(9) = -0.487, p = 0.638) and PBI (t(9) = -0.218, p = 0.832).

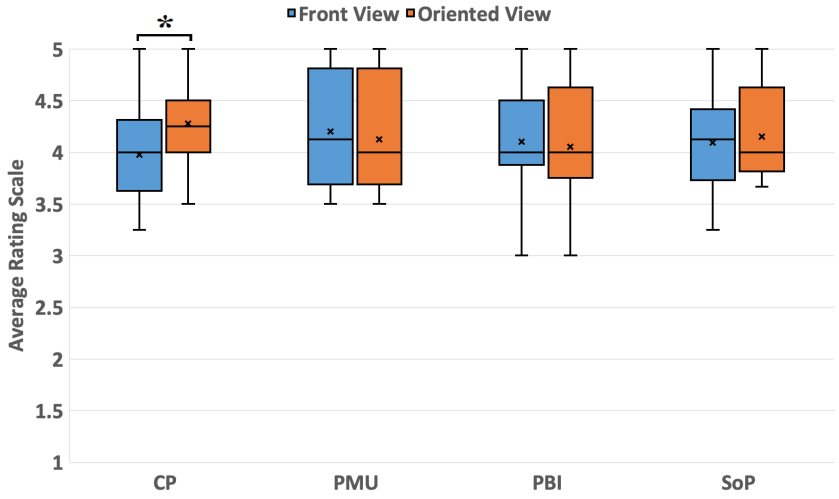


FIGURE 3.10: Results of the Social Presence questionnaire (CP: Co-presence, PMU: Perceived Message Understanding, PBI: Perceived Behavioral Interdependence, and SoP: Overall Social Presence;∗: statistically significant difference)

After trying both view interfaces, participants were asked to choose which interface

they thought was the best for the LEGO assembling task in the study. As shown in Figure 3.11, of the 10 participants recruited to the study, five participants preferred to use Front View mode, and five participants preferred to use Oriented View mode. A chi-square goodness-of-fit test indicated that the two view modes were equally preferred by the participants ($\chi^2(2) = 0.000$, p = 1.000).

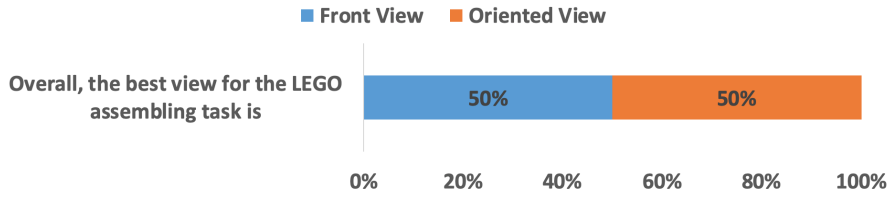


FIGURE 3.11: User preference about which interface they preferred best for the LEGO assembling task

Based on the rating of usability and social presence questionnaire, rendering the local single-frame point cloud scene as an oriented view did not show any significant difference compared to a front view in overall social presence in remote collaboration. However, participants felt their partners were significantly obvious to them and could easily catch their attention (Co-presence) while using the Oriented View mode. This might be because all the items in this study were located close to each other in a small area and could be captured in the same scene without the local worker turning his or her head very much. In this case, the remote expert could easily find items he/she wanted in the local worker's scene. There was almost no need for the remote expert to imagine the spatial relationship between items since it was straightforward in the shared view. Therefore, we did not find a significant difference in overall social presence while presenting the local spatial layout with additional orientation information.

Based on the feedback of the post-experiment questions, we believed that both interfaces had their own advantages and disadvantages. In terms of Oriented View sharing, some participants mentioned that they felt they could better understand where their partners were watching, and felt more closely connected with the partner not only spatially but also mentally (Co-presence). Moreover, they also felt less dizzy and claimed that this view method provided a more immersive and realistic MR user experience. However, some participants also felt that they sometimes failed to follow the worker's frequent or fast movements since the Oriented View sharing interface was physically stressful. They became too tired to pay enough attention during communication while guiding the local worker. This is also why we found a significant difference in terms of physical stress in the rating scale question.

In contrast, participants felt that the Front View sharing method was less physically stressful and more comfortable to focus on the task itself. However, participants claimed that without the spatial orientation cues, they could quickly lose spatial direction, which was also very difficult to recover. They also felt sick when their partners constantly moved in a way that they did not expect. We observed that although the Front View was used in the test, the participants still oriented their heads based on the workers' head movements, and kept changing their head pose during the tasks, which was not necessary at all with this interface.

3.3 Conclusion

In this chapter, I presented a prototype remote guiding system that we developed using VR headsets combined with depth sensors. Both the local worker's workspace and the remote expert's hand gestures were captured by depth sensors and reconstructed as 3D point clouds in the same VR scene. We compared two different view-sharing modes for the remote expert, Oriented View and Front View sharing.

The results of the study showed that Oriented View mode increased the experts' physical stress since it required the experts to follow the local workers' view orientation in order to catch the current view. Participants also felt the local workers' actions attracted their attention while using the Oriented View more than the Front view (Co-presence). However, based on subjective feedback, the Oriented View interface might cause the experts to feel fatigued after long-term usage, and could easily lead them to lose their partners' view. On the other hand, while using the Front View interface, the experts could pay more attention to the task itself, rather than matching the worker's view. We also noticed that the experts usually did not need to imagine the spatial relationship between items because they could always be captured together in the same scene in this study.

In the next chapter, I extended the prototype system into a larger scale workspace, such as a room-size workspace. At this scale, items may be located in different places far apart from each other, and cannot be captured together in the same scene. In this case, helping the remote expert to understand the spatial relationship between different items may significantly increase the efficiency of the remote collaboration tasks.

Chapter 4

Room-Scale Mixed Reality Remote Collaboration System

To support remote collaboration in a room-scale workspace, we developed a prototype system that reconstructed the local physical environment and shared it as a 3D VR scene with the remote expert. Using a VR HMD, the remote expert could then freely navigate himself/herself through the virtual copy of the local worker's real environment with a better understanding of the spatial relationships between objects in the local workspace. In this case, the remote expert might feel as though they were sharing the same workspace as the local worker.

Our remote collaboration system captured and reconstructed the local scene as a static 3D point cloud set. Once created, there was no real-time update of the point cloud from the local worker's side. However, we developed several different interface ideas to provide real-time feedback to show the local worker's actions.

Overall, our MR based remote collaboration system aimed to fulfill the following room-scale design requirements:

- A shared view in order to build common ground;
- Immersive experience and spatial awareness for the remote expert to understand the local physical layout;
- Independent viewpoint control for the remote expert to explore the view of the local space independently;
- Real-time feedback for the remote expert to check the local worker's actions;

- Different kinds of guidance cues for the remote expert to guide the local worker;
- Approaches to show remote guidance on the local side;
- Independent viewpoint control for the local worker to observe the remote guidance and work on the task;

4.1 System Overview

We captured the local user's surroundings as one integrated 3D point cloud and displayed it as static VR content in the remote expert's VR headset to build common ground. Based on this immersive and spatial experience, the remote expert could feel like they were virtually placed in the same workspace as the local worker. With accurate position tracking enabled, users on both sides were able to share the movement and view direction of themselves independently in this shared VR workspace, which produced a natural collaborative experience similar to as if they were face-to-face. We intended to use single-frame 2D/3D live views from the local side to show the real-time changes. Visual cues and voice contact were also enabled to support natural communication. In addition, the local worker had the video see-through AR view to observe the remote guidance cues and interact with the objects in the local physical world.

Our prototype system was subdivided into two sub-systems: (1) the local capturing and sharing system that supported the worker on task completion, and (2) the remote guiding and viewing system that enabled the expert to provide real-time help (Figure 4.1).

FIGURE 4.1: The static local environment capturing and sharing with real-time feedback for MR remote collaboration. A: the local user stood in front of a workspace, identifying a particular LEGO model which the remote expert pointed to. B: The remote expert observed the local environment and guided the local worker by pointing in the VR world.

In the following sections, we discuss the hardware and software setup and implementation of these two sub-systems. The main functions of our remote collaboration system were:

- Being able to capture and share the local environment as a static 3D point cloud scene, with 2D/3D live view to show the real-time changes;
- Presenting the local view in VR for the remote expert;
- Supporting natural communication cues such as pointing and speech;
- Using a video see-through device to provide an AR guide for the local worker.

4.2 The Local Workspace Setup

The local system was responsible for capturing the local scene and providing an AR view for the local worker observing their surrounding workspace. To achieve this requirement, the local user wore a VR headset (HTC Vive [1]) during the task process. One depth sensor (Intel RealSense R200 [2] with an operating range from 0.5m to 3.5m) was attached to the front face of the headset, with its RGB camera located in the middle of the headset surface facing toward the workspace (Figure 4.2). The captured color video stream was then passed through the headset display to create a video see-through AR view, allowing the local worker to directly see the surrounding environment and freely move in the physical world.

FIGURE 4.2: The local worker's video see-through view and VR headset setup

[1] ttps://www.vive.com/
[2] https://software.intel.com/en-us/articles/realsense-r200-camera

For each frame, the depth data captured was firstly aligned with the RGB data to create a point cloud of the 3D space. The Intel RealSense SDK library [3] enabled the Unity engine to grab the point cloud from their sensor directly. Therefore, instead of calculating the point cloud by using the method we introduced in Section 3.1, we called the library functions directly to build the point cloud for each frame.

While the local worker was walking around in the local workspace, the system captured a set of point clouds from different locations. These single frame point clouds were then fused together into one integrated model to copy the entire larger-scale local physical environment in the VR world. This process was described in more detail in section 4.4. By streaming this integrated point cloud model to the remote side, the remote expert could check the local work environment by viewing it with a VR display.

The Vive Lighthouse tracking achieved an expected accuracy of about 2mm with the worst-case latency of 1ms for head tracking [91]. Based on this, we could collect the local worker's head position and orientation information in real-time. This information was then streamed to the remote side. Based on this information, we could render the local worker's view frustum in the remote expert's VR environment to provide a viewpoint awareness cue (see Figure 4.3). In this case, the remote expert had the ability to make visual contact with the local worker, which could increase the remote expert's mutual awareness.

On the local side, we used a VR-Ready PC set up with an Intel Core i7, 8GB RAM, and NVIDIA GeForce GTX 970 GPU, running Windows 10. This local PC was responsible for the local data processing by using the Unity game engine, such as rendering the local video see-through view and the remote guidance, integrating and streaming the local 2D and 3D views to the remote side. The HMD was directly connected to the PC via USB and HDMI cables, while the depth sensor and RGB camera were connected via a USB cable.

4.3 Remote Workspace Setup

The remote system was responsible for displaying different user interfaces to help the remote expert provide guidance on the tasks. In order to view the interfaces, the remote expert was also asked to wear a VR headset (the HTC Vive) during the guiding task, which enabled him/her to view the shared 3D local scene and freely move around in the 3D VR world while navigating himself to the target location. In this way, the

[3] https://unity3d.com/partners/intel/realsense

remote expert could quickly identify any target objects while guiding the local worker. Figure 4.3 shows an example of the remote expert's view.

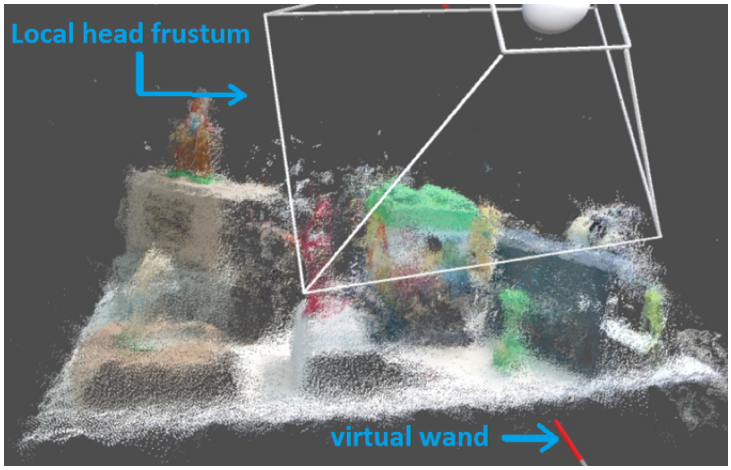


FIGURE 4.3: The remote expert's view, showing the view frustum of the local user (white wireframe of the frustum)

The static 3D point cloud capture of the local scene provided the remote expert with an independent view while the local worker was working in a large (e.g., room-scale) workspace. At the same time, our system also streamed the local worker's current view frame to the remote expert to act as a reference for the remote expert to observe real-time changes in the local workspace. This current view frame was rendered as either a 2D window view or a 3D point cloud view based on different interface design purposes, discussed in detail in Chapter 6.

The local worker's view frustum was also shown in the remote expert's VR scene (Figure 4.3), allowing the remote expert to see the local worker's head movement and view direction, which enabled the expert to check whether or not the worker was following their guidance. The expert also held one Vive handheld controller, which was rendered as a virtual wand in the VR world, and was streamed back to the local side as one virtual cue overlaid on top of the worker's view (see Figure 4.2 and Figure 4.3). In this way, the remote expert could provide virtual pointing feedback to help guide the local worker.

The remote headset was wirelessly connected to a PC (called "Remote PC") by using the TPCast Vive wireless adapter [4]. This Remote PC was set up with an Intel Core i5, 8GB RAM, and NVIDIA GeForce GTX 970 GPU, running the Windows 10 operating system. The remote scene was also rendered with the Unity game engine.

[4]https://www.tpcastvr.com/product-vive

4.4 Scene Capture

We developed a simple method for creating the final 3D point cloud. In order to capture and render the entire local physical scene as one single dense point cloud model, we attached an Intel RealSense R200 sensor to the front face of the VR headset, facing toward the workspace (Figure 4.2). While the local worker was walking around the local workspace, the system could capture the current view based on the aligned RGB frame and depth frame from the sensor. Each pixel of this view was then projected into the Vive VR camera coordinate system using the sensor's intrinsic parameters, which turned the view into a dense point cloud. The position of the Vive VR camera, taken to be the same as the Vive headset, was captured by the Vive lighthouse hardware. In this case, the point cloud of each frame could be finally mapped into the Vive VR world coordinate system.

We used a keyframe based registration method to do the local scene capturing. The scene reconstruction process started by the local worker manually pressing the trigger on the Vive controller. While the scene capturing was running, the first frame that was captured was considered the initial keyframe of the system. The point cloud of each following frame was then identified as a keyframe or not based on its relevant position to the previous keyframes. If the current point cloud had no more than 20% to 40% of its points overlapping with the previous three keyframes, it was chosen as the next keyframe. If it was a keyframe, this point cloud data would be saved and stitched with previous keyframes by using the Iterative Closest Point (ICP) algorithm [19]. If it was not, we just deleted the point cloud data of this frame. While the worker was walking around the local workspace, the system kept adding new keyframes onto the previous ones. After the entire local workspace was captured (as judged by the local worker), the local worker could press the trigger again on the Vive controller, and the system would stop collecting new keyframes. Figure 4.4 shows the flow chart of the entire capturing and stitching process [5].

The size of the local workspace that could be captured was based on the Vive Lighthouse tracking area (usually no more than 3.5m*3.5m). During the scene capture, the local worker needed to move slowly through the workspace. Otherwise, if the worker moved too fast, two sequential frames might not have overlapped areas, which led to failed keyframe registration. In order to achieve real-time scene capture, we set the resolution of the color and depth map as 320 * 240 pixels. In this case, the point cloud from each frame only contained less than 320*240 points (noise points were removed). A

[5] Video link of scene capture: https://www.youtube.com/watch?v=6lPSQeKRo50

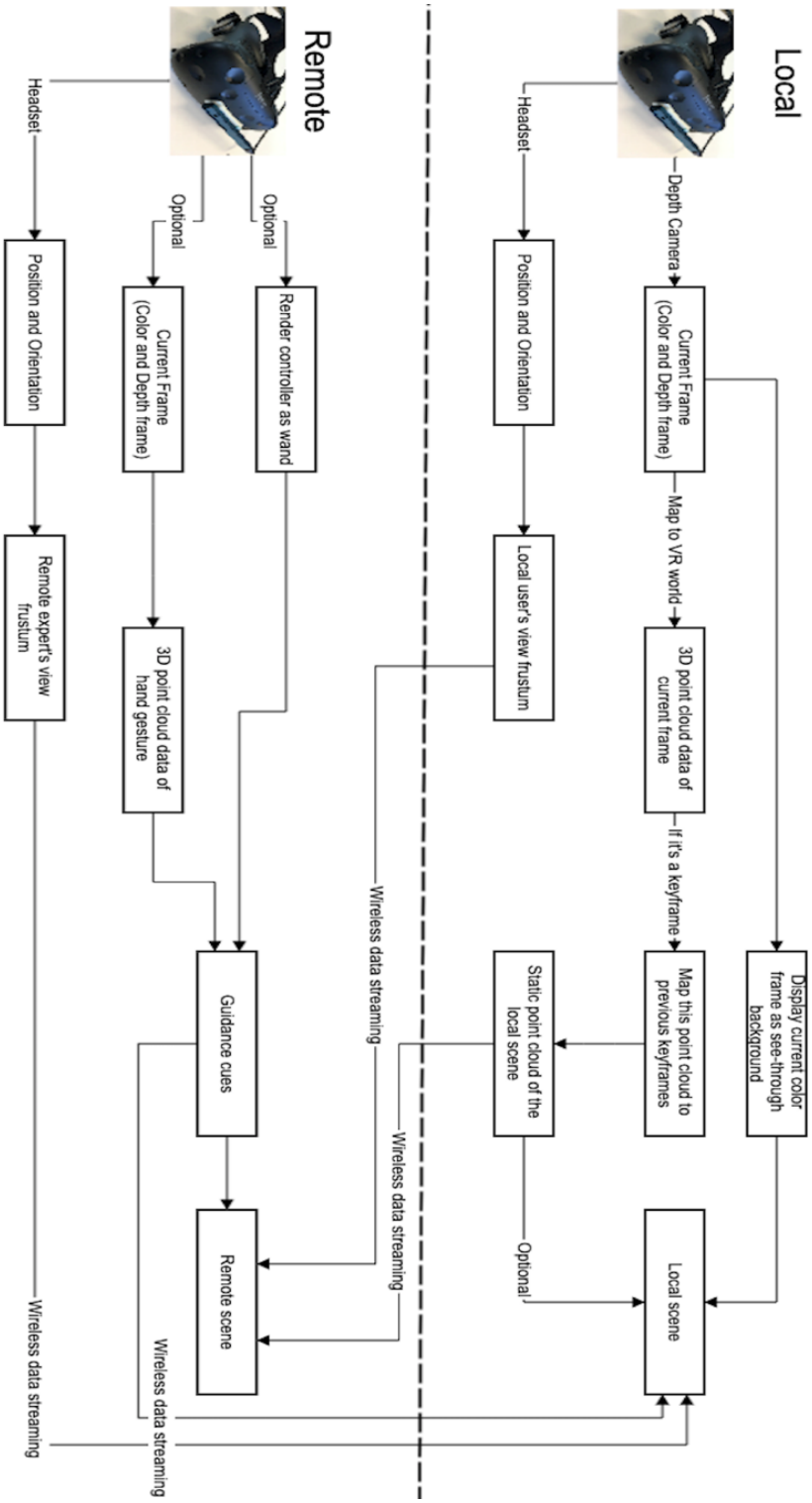


FIGURE 4.4: The flowchart of the static environment capturing and sharing system

workspace of 3.5m * 3.5m, usually needed 30 to 50 keyframes captured to copy the entire scene, which took around 30 to 60 seconds for the local worker to finish the capture process. Since we restricted the frames from the depth sensor with a low resolution, the reconstructed 3D point cloud set could only present the general geometry of the local workspace. Figure 4.5 shows an example of the keyframe registration.

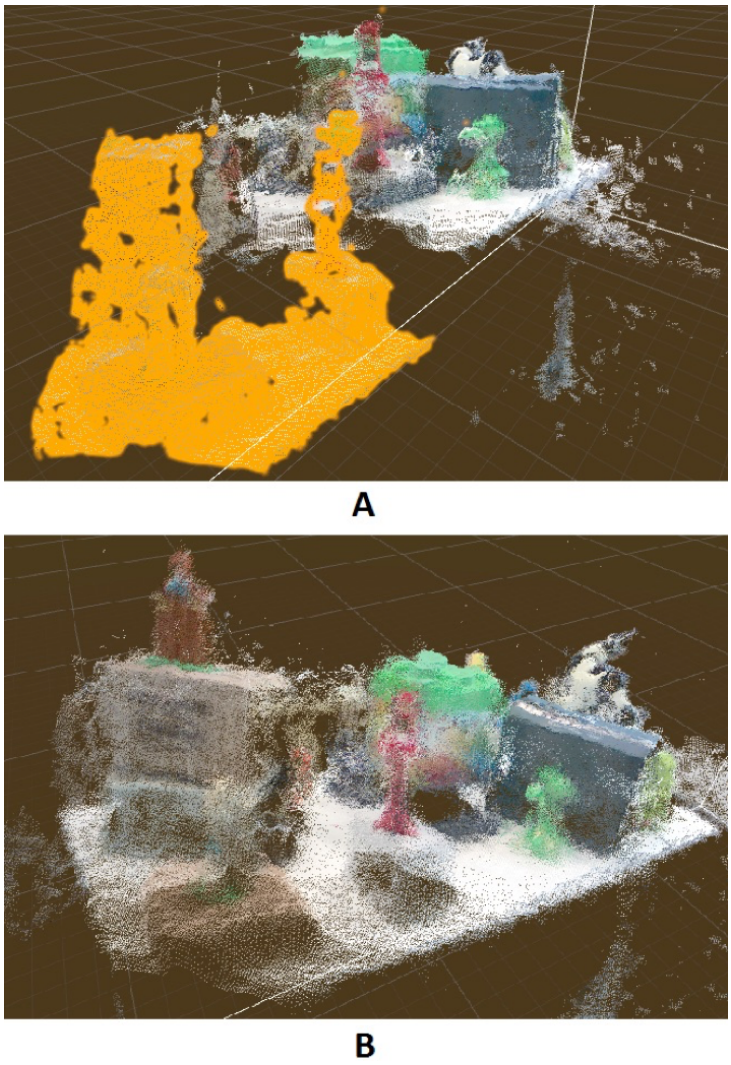


FIGURE 4.5: Keyframe registration. A: The yellow point cloud is built based on a new frame; B: After the registration, the new frame is stitched to the previous ones

During the scene capture process, once a frame was detected as one keyframe, the system automatically sent the point cloud data of this frame to the remote side. On the remote side, since each point cloud data set was received as the keyframe, the remote system just kept rendering a new point cloud into the VR scene. The remote expert wearing the VR headset would see the point cloud updating itself in the VR world to show the scanned local physical environment until the local worker finished the scene capture

process.

On the local side, the scene capture process could reach 15 fps during keyframe registration. On the remote side, since we streamed each keyframe (no more than 320 * 240 points) separately, the remote system did not need to render a large point cloud set at once. Therefore, the remote system kept running at 45 fps during the entire process.

4.5 Mutual Awareness and Remote Guidance

Using the Vive lighthouse tracking on the HMD, we could collect the local worker's head position and orientation information in real-time. By streaming this information to the remote side, the local worker's view frustum could be rendered in the remote expert's VR space (Figure 4.3). In this case, the remote expert could observe the local worker's head movement and view direction, which enabled the expert to check whether or not the worker was following the guidance given. Our system also could stream and render the remote expert's view frustum in the local worker's AR view. Since the remote expert mainly kept watching the target area, the local worker could quickly locate the target by searching for the remote expert's view frustum in the shared AR scene.

Seeing the real-time head frustum from the partner's side for both local and remote users might increase the efficiency of communication between each of them. Besides supporting the awareness of view position and orientation, the expert was also asked to hold one Vive controller which was rendered as a wand in the VR world and was streamed back to the local side as a virtual cue overlaid on top of the worker's view (Figure 4.2). Moreover, we also tried to capture the expert's real hand in the 3D space (using the method introduced in Section 3.1) to provide a more natural communication cue. In this way, the remote expert could provide virtual pointing feedback or more complicated guidance gestures to help the local worker. Furthermore, the system sent both users' speech from one side to the other. Therefore, during the collaborative task, the remote expert could provide pointing guidance by using the controller and/or directly talking with the local worker.

4.6 Data Streaming and Rendering

The local scene was captured and rendered as a dense point cloud on the local side, and then, it was sent to the remote side before the task started. During the task, the system streamed the local worker's headset position, along with the current view captured by the sensor, to the Remote PC to assist the expert. At the same time, the Remote PC sent

the virtual wand information as guidance to the Local PC. All of this data streaming was based on a wireless connection between the Local PC and the Remote PC. In order to achieve real-time data communication, we used the NETGEAR Nighthawk X6 WiFi Router and the sharing service supported by HoloToolkit [6].

The Unity game engine [7] provided excellent support for VR scene rendering while using the HTC Vive headset. All of the point cloud data was rendered as vertical meshes in Unity. On the local side, the frame rate reached 15 fps while reconstructing the local scene before the task and 30 fps during the task with real-time single frame streaming enabled. On the remote side, the frame rate reached 45 fps during the tasks.

4.7 Conclusion

In this chapter, I describe the basic setup of our MR remote collaboration system. Based on this system setup, we tried to explore the answer to our research questions by conducting four different user studies:

- Compare the static scene capturing and sharing, with single frame point cloud sharing (Oriented View mode introduced in Chapter 3);
- Compare different representations of the view mediums while using our static scene capturing and sharing remote collaboration system;
- Analyze user behaviors based on two user studies while using our remote collaboration system.

In the following chapters, I describe these studies in more detail. Based on these studies' results, I present our solution for extending remote collaboration from small workspace to room-scale workspace with MR features enabled.

[6] https://github.com/Microsoft/MixedRealityToolkit-Unity
[7] https://unity3d.com/

Chapter 5

Static Visual Representation

In this study, we evaluated our MR system, which supported the capture of the entire local physical work environment for remote collaboration in a large-scale workspace. By integrating the keyframes captured with the external depth sensor into one single 3D point cloud data set, our system could reconstruct the entire local physical workspace into the VR world. In this case, the remote expert could observe the local scene independently from the local user's current head and camera position, and provide gesture guiding information even before the local user was looking at the target object. We conducted a user study to evaluate our system's usability by comparing it with our previous Oriented View system (Figure 3.8), which only shared the current single frame camera view together with the real-time head orientation data. Our results indicated that this entire scene capture and sharing system could significantly increase the remote expert's task performance in terms of target searching in a large-scale workspace with less mental stress compared to our previous system. Furthermore, based on subjective feedback, we also tried to summarize the advantages and disadvantages of our MR remote collaboration system.

5.1 Experiment Setup

Our remote collaboration system (Figure 5.1) captured and reconstructed the local scene as a static 3D point cloud set. Once created, there was no real-time update of the point cloud from the local user's side. By comparing with a single frame sharing system (including real-time feedback from the local side), we conducted an initial user study to investigate the usability and social presence of our static local environment capturing and sharing system. We hypothesis that capturing and sharing the entire room-scale local scene as a 3D replica could help the remote experts to

- decrease the objects searching time,
- enhance the understanding of the local physical layout, and
- provide efficient guiding support.

FIGURE 5.1: The static local environment capturing system for MR remote collaboration. A: the local worker stood in front of a workspace, identifying a particular LEGO model that the remote expert was pointing at. B: the 3D point cloud of the current frame from the local side, remote guiding gestures, and the remote expert's view frustum were displayed in the local worker's VR headset. C: the remote expert observed the reconstructed local environment in the VR world and guided the local worker by using hand gestures. D: the entire scene of the local workspace had been mapped as one integrated 3D point cloud and rendered in the remote expert's VR headset.

Since our system focused on the remote experts' experience, participants were recruited for the role of the remote expert. Ten people took part in the study, six men and four women, aged 24 to 33 years old. All of them had previous experience with 2D video streaming interfaces, such as Skype and Facebook Live, but with limited VR and AR experience. Only two participants had tried a remote collaboration AR or VR system before. Each participant was asked to guide a worker to find some sets of LEGO models in the local workspace by using two different interfaces: (1) our static scene capturing and sharing interface and (2) one oriented view interface, which is the same as we used in the study of Chapter 3 with only the current single frame provided [1]. The task was considered finished once the local worker found all the LEGO models correctly.

We had seven different LEGO models. Each of them had one unique color (red, orange, green, blue, yellow, white, or brown), and was placed at a distance from all others in a large local workspace. In this case, these LEGO models could be found by the users based on their color, and they could not be captured together by the sensor in one frame. For each interface, participants performing as the remote experts were given a random order of the LEGO models based on the models' colors before the task. After the task

[1] Video link of two interfaces: https://www.youtube.com/watch?v=Ec1120oQUx4

started, remote experts were asked to guide the local workers to find the models one by one based on the pre-defined order. The first model searching trial was considered a training trial, and the next six searching trials were considered formal trials. The two interfaces were used in a different order for each participant to exclude potential learning effects.

For the scene capturing and sharing interface, the system reconstructed the local workspace into a 3D point cloud before the task started, and then streamed the point cloud data to the remote side. During the task, both users could see the partner's view frustum in the VR world, which provided a straightforward way for them to check their partner's focus in the scene. Figure 5.2 and Figure 5.3 show examples of the local and remote users' view. They were almost the same. One static point cloud set showed the local surrounding environment, one view frustum showed the partner's current focus, and one dynamic point cloud set showed the remote gesture guidance.

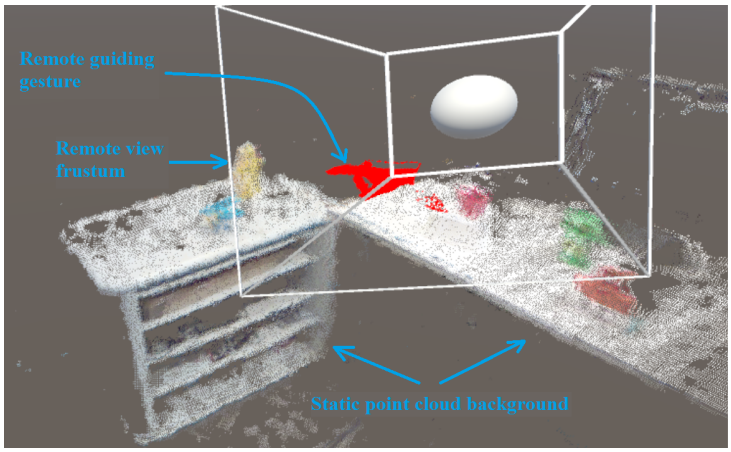


FIGURE 5.2: The local worker's view interface: remote view frustum and guiding gesture were used to show the expert's focus and guiding cues

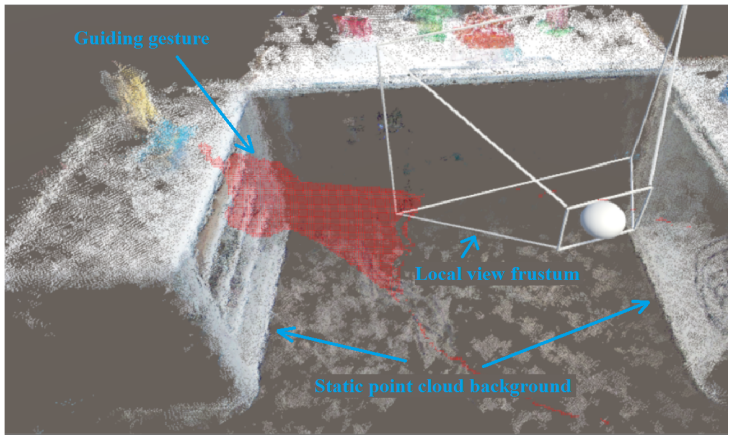


FIGURE 5.3: The remote expert's view interface: local view frustum was used to show the worker's focus, and the expert could see his/her own hand gestures in the VR world

For the oriented view interface, the system only captured and mapped the current frame into one 3D point cloud (Chapter 3, Figure 3.8). By sharing the local worker's head position and orientation, this single frame point cloud could be mapped to the same position as it in the real world. Therefore, if the remote expert wanted to catch up with the current local view, he or she just needed to check the worker's view direction in the shared VR space. For both interfaces, participants could use hand gestures and speech to guide the local worker.

We used a within-subjects design. Participants recruited were asked to use both view interface for the LEGO model searching tasks. The time taken was automatically measured and recorded by the system. At the beginning of the user study, participants were given a general explanation about the features of the two interfaces and the procedures of the tasks in detail. After this, they were asked to wear the headset to start the study. After the participants completed each condition, they were asked to provide some feedback by answering the usability and social presence related interview questionnaire using a seven-point Likert scale (1 to 7 with 1 indicating strongly disagree while 7 indicating strongly agree). In addition, we also asked participants to provide some comments in response to questions in the post-experiment questionnaire.

5.2 Experiment Result

A paired-samples t-test was used to determine whether there was a statistically significant mean difference in task completion time between the two interfaces. Data are mean $\pm$ standard deviation, unless otherwise stated. The assumption of normality was not violated, as assessed by Shapiro-Wilk's test (p = 0.793). As shown in Figure 5.4, participants spent less time on completing the model searching task while using the scene capturing and sharing interface (mean = 67.750, standard deviation = 17.145) seconds as opposed to the oriented view interface (mean = 92.050, standard deviation = 16.049) seconds, a statistically significant decrease of 24.3 (95% CI, -41.374 to -7.226) seconds, t(9) = -3.220, p = 0.010.

The results for usability custom rating items are shown in Figure 5.5. Using a Wilcoxon signed-rank test, we found that participants significantly felt more confident while using the oriented view interface (median = 7.0, interquartile range = 1.000) compared to the scene capturing and sharing interface (median = 6.0, interquartile range = 1.250) (z = -2.111, p = 0.035). We also found that there was statistically significantly increase of mental stress while using the oriented view interface (oriented view: median = 2.0,

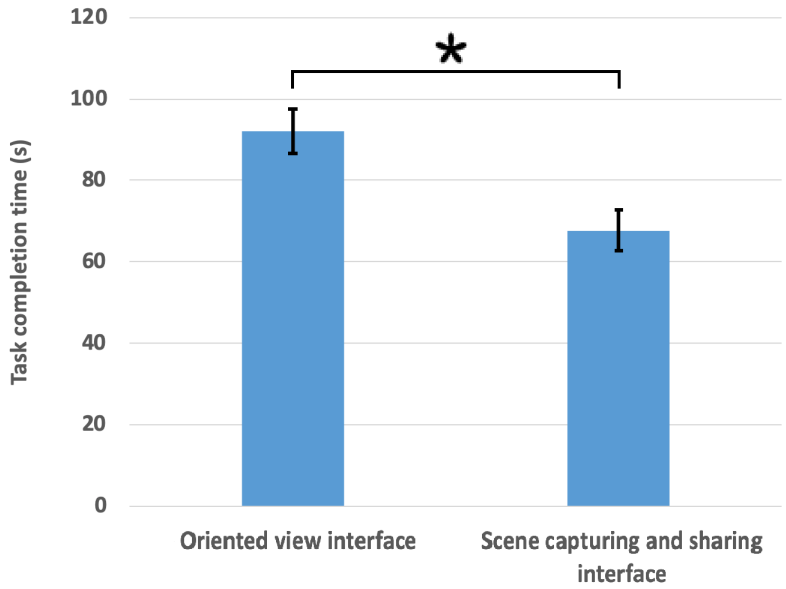


FIGURE 5.4: The task completion time(∗: statistically significant difference)

interquartile range = 1.000; scene capturing and sharing view: median = 1.5, interquartile range = 1.000; z = -1.983, p = 0.047). No significant difference was found for the rest usability rating questions (Like to use: z = 0.259, p = 0.796; Easy to use: z = 0.921, p = 0.357; Helpful: z = 1.265, p = 0.206; Physical stress: z = -0.333, p = 0.739).

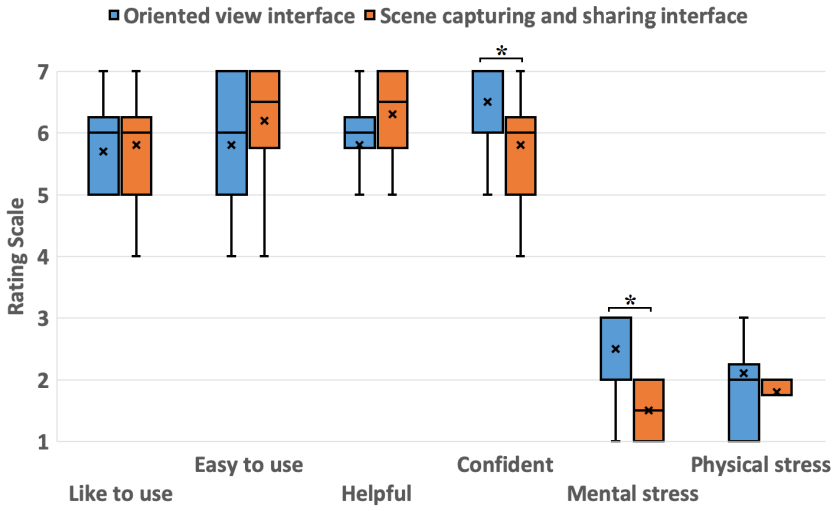


FIGURE 5.5: Results of the usability rating scale (∗: statistically significant difference)

We used two sub-factors from the social presence questionnaire (SoPQ) [48], which included Co-presence (CP) and Perceived Message Understanding (PMU). Each of these sub-factors consisted of six closely interrelated questions scaled from 1 (strongly disagree) to 7 (strongly agree), which represented a group of Likert scale measurements. To analyze Likert scale data, we first calculated a composite score from the six questions

for each sub-factor [110], and then a paired-samples t-test was used to determine whether there was a statistically significant mean difference between the two view modes. As shown in Figure 5.6, we found no significant difference in term of social presence (t(9) = 0.079, p = 0.939). We also analysed each sub-factors, and no significant difference was found regarding CP (t(9) = 0.583, p = 0.574), nor on PMU (t(9) = -0.467, p = 0.651).

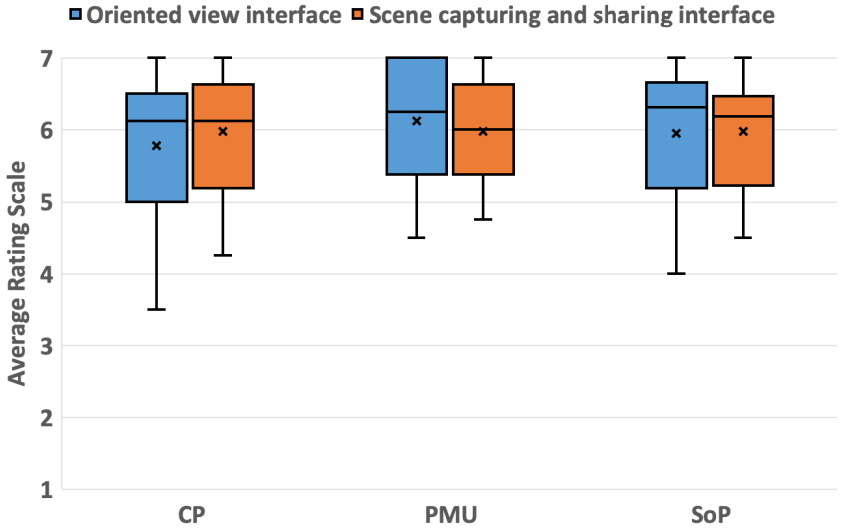


FIGURE 5.6: Results of the Social Presence questionnaire (CP: Co-presence, PMU: Perceived Message Understanding, and SoP: Overall Social Presence)

In addition, we also asked the participants to rate on a 7-point scale (1: strongly disagree – 7: strongly agree) on how much they agree with the statement "This interface is useful for learning the local environment", "This interface is useful for finding the targets" and "This interface is useful for guiding the local partner" for both two view interfaces. Using a Wilcoxon signed-rank test, the results (Figure 5.7) showed that participants significantly preferred to use the scene capturing and sharing interface on target searching (oriented view: median = 5.0, interquartile range = 2.000; scene capturing and sharing view: median = 6.0, interquartile range = 1.000; z = 2.326, p = 0.020). No significant difference was found regarding the impact on learning the local environment (z = 1.852, p = 0.064), nor on the impact on guiding the local partner (z = 1.134, p = 0.257).

5.3 Discussion

While using the scene capture and sharing interface, most of the participants pointed out that the reconstructed 3D point cloud set of the local work environment provided them with a direct view of the entire workspace, making it easier to find the target object. The task used in our study was mainly a searching task. Based on the results, we knew that participants spent a significantly short time while using the scene capturing and sharing

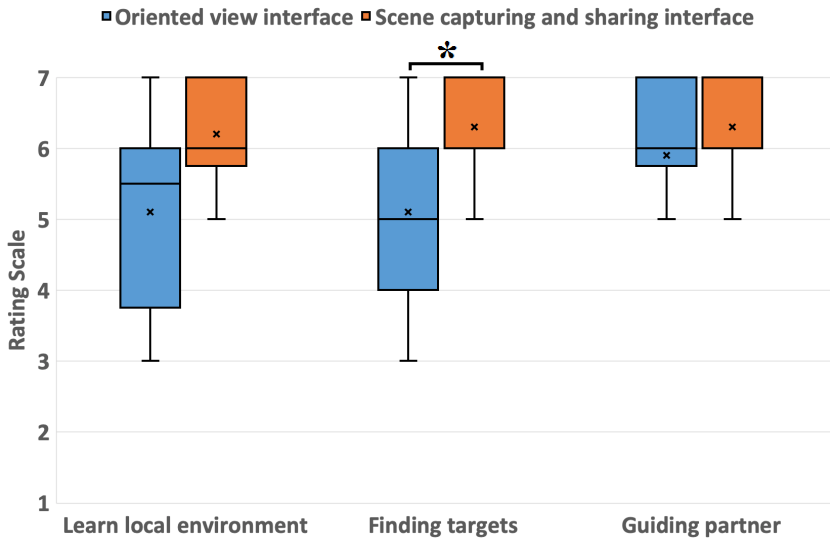


FIGURE 5.7: Subjective rating on user experience(∗: statistically significant difference)

interface on item searching, which verified our assumption that capturing and sharing the entire room-scale local scene as a 3D replica could decrease the object searching time. However, the subjective rating feedback did not show significant differences between the two interfaces for the impact on learning the local environment's layout and guiding the remote partner. Participants reported that seeing the view frustum of their partner helped them figure out their partners' current focus areas in the scene. However, there was no real-time feedback during the task process to inform them if their partners had found the right object or not because the 3D virtual scene was captured before the task started and would not be updated during it. Furthermore, participants also mentioned that using the hands to guide the local worker was natural; however, they could only use gestures in a small field just in front of them. Otherwise, the system would be failed to capture the hands. The depth sensor we used to capture the users' hand gestures only supported a small field of view, which limited the detection range.

In contrast, the oriented view interface supported current frame capturing and sharing in real-time; therefore, participants said it was easier for them to control the task process by seeing the partners' actions while using this interface. In this case, they felt more confident in completing the object searching task (Figure 5.5). However, this interface only shared one frame at a time, which meant that the participants took more time to find the target objects. Participants often used words like *"look at your left side"*, *"turn to your right side"* or *"back to front"* to ask the local workers to scan the local work environment first before they could actually start the guidance. A few participants also pointed out that if their own viewpoints were different from the local workers, they might see gaps

in the 3D point cloud, making it challenging to identify the targets. This problem was partly due to the occlusion caused by objects in the foreground blocking the view of the local depth sensor.

Overall, each interface had its own advantages and disadvantages. Static scene capturing and sharing could provide an overview of the large local workspace for the experts to quickly locate the target position but without real-time feedback from the local side. Therefore, the remote experts needed to pay more attention to whether or not the local worker was following their guidance. On the other hand, the oriented view interface could show a real-time view of the local workspace, but it was limited by the depth sensor's view angle and detection range, so this single frame view was presented in a low resolution with little details. Therefore, participants performing as remote experts needed to spend more time on target searching.

5.4 Conclusion

In this study, I have evaluated our prototype remote collaboration system that captured and shared the entire local workspace as a single dense point cloud in the VR environment for the remote expert to have an independent viewpoint control. To investigate the usability of our prototype system, we compared it with a single-frame oriented view interface.

The results indicated that the participants were able to complete the object searching task much faster with the scene capture system than the oriented view interface. However, without real-time feedback, the remote expert could not identify if the worker following the right guidance. In the next user study, I extended the current static scene capture and sharing system to support real-time feedback from the local side. The local physical work environment would be captured and mapped as a static 3D dense point cloud, and at the same time, the system would also support the local live view to show real-time changes. In this case, the remote expert could have the ability to monitor the local working process.

Chapter 6

Static Visual Representation with Live Feedback

In this user study, I present a prototype MR system with a combination of different view media to support remote collaboration between a local worker and a remote expert in a large-scale workspace. By combining a low-resolution 3D point cloud of the environment surrounding the local worker with a high-resolution real-time view of small focused details, the remote expert could see a virtual copy of the local workspace with independent viewpoint control. Meanwhile, the expert could also check the current actions of the local worker through a real-time feedback view. We intended to evaluate the following research questions:

- How do different representations of the view media affect the task performance of remote collaboration?
- How do different representations of the view media affect the user experience, including usability and social presence?

We conducted a pilot study followed by a formal study to evaluate our system by comparing the performance of three different interface designs, showing the real-time view in forms of a 2D first-person view, a 2D third-person view, and a 3D point cloud view. We found no difference in average task performance time between the three interfaces, but there was a difference in user preference. Each interface had its own advantages and limitations, for example, the 2D first-person view was easy to monitor the process, the 2D third-person view provided better co-presence experience, and the 3D point cloud view was more straightforward for observing the local changes.

6.1 Pilot User Study

In order to evaluate the usability of our prototype system, we conducted a pilot study comparing different real-time feedback approaches for the remote expert. This study aimed to explore how different real-time visual representations of the local worker's space could affect the remote user's performance while the local worker was working in a large workspace.

6.1.1 Interface Design

Our study mainly investigated the remote expert's user experience with different spatial sharing technologies. We created a hybrid interface that combined a low-resolution 3D point cloud with a high-resolution real-time view for small focused details. The advantage of the hybrid interface was that it provided large-scale static 3D information simultaneously as real-time 2D or 3D detailed information.

Based on our current system setup, we could share two types of spatial information from the local worker with the remote expert:

- A broad view of the surrounding background in a 3D virtual point cloud reconstruction of the local environment.
- A small detailed foreground view with real-time feedback in terms of a 2D video or a 3D point cloud based on the local worker's current view.

We created three interfaces to investigate our system [1]. Each of them had a 3D point cloud background enabled, but with different foreground display approaches:

1. First-person view (FPV): A static 3D virtual point cloud of the local scene was displayed as a background in the remote expert's VR world. Real-time 2D video of the local worker's view was displayed at the top-right corner of the remote expert's view as a 2D window, which always followed the remote expert's head movement (see Figure 6.1 A).
2. Third-person view (TPV): A static 3D virtual point cloud of the local scene was displayed as a background in the remote expert's VR world. Real-time 2D video of the local worker's view was displayed as a 2D window and attached to the local worker's head view frustum in the remote expert's VR world (see Figure 6.1 B).

[1] Video link of three interfaces: https://www.youtube.com/watch?v=fFsp7A9z1TQ

3. Point cloud view (PCV): A static 3D virtual point cloud of the local scene was displayed as a background in the remote expert's VR world. The current frame of the local worker's view was captured by a short-range depth sensor and rendered as a 3D point cloud in the VR world to show real-time feedback from the local side to the remote side (see Figure 6.1 C). Since the point cloud of the current view was directly overlaid on top of the static point cloud set of the local scene, the remote expert could directly see the current changes in his/her 3D VR space.

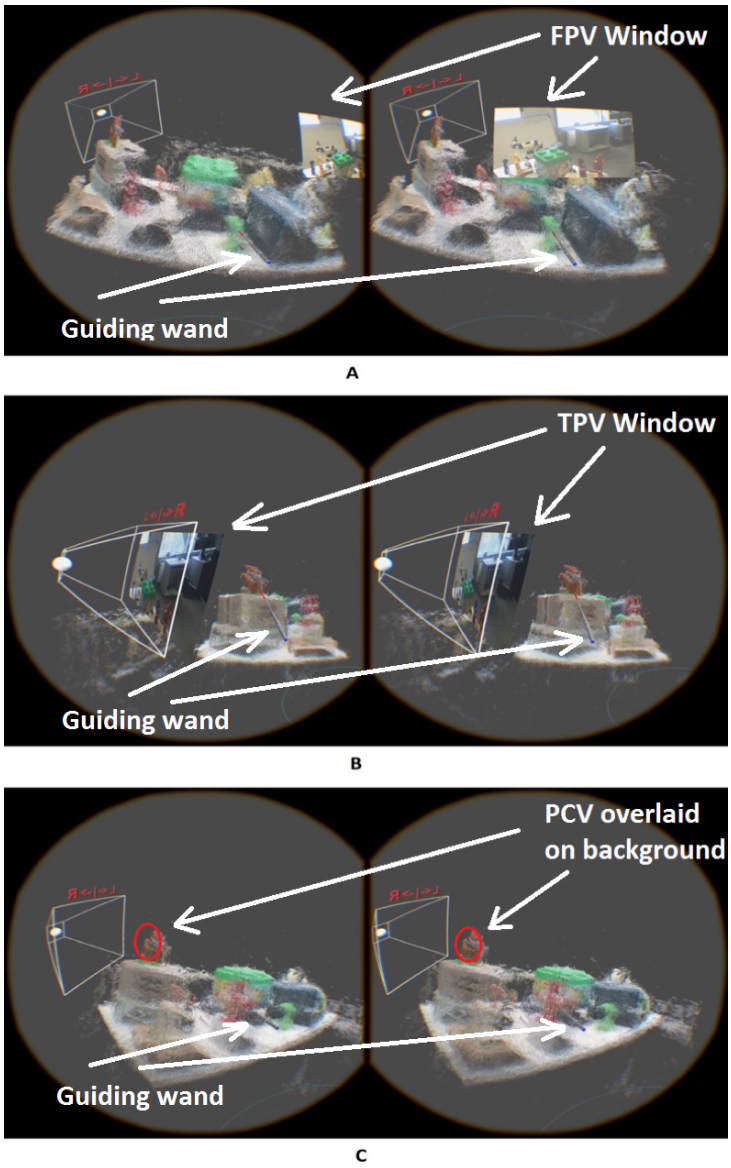


FIGURE 6.1: Three interfaces. A: 3D point cloud background with 2D video foreground in first person view (FPV); B: 3D point cloud background with 2D video foreground in third person view (TPV); C: 3D point cloud background with 3D point cloud foreground (PCV)

To achieve the above interface setup, we modified the design of our local worker's VR

headset. As shown in Figure 6.2, we used a long-range depth sensor (Intel RealSense R200 with an operating range from 0.5m to 3.5m) to capture and rebuild the local scene as the background view. At the same time, one high-resolution USB camera (with the support of 1080p and 60fps) was used to capture the real-time 2D video of the local workspace, and one short-range depth sensor (Intel RealSense SR300 with an operating range from 0.2m to 1.5m) was used to capture the real-time 3D point cloud of the local workspace. The two views from the USB camera and short-range depth sensor were chosen to display as the foreground view based on the different requirements of each interface design during the experimental tasks.

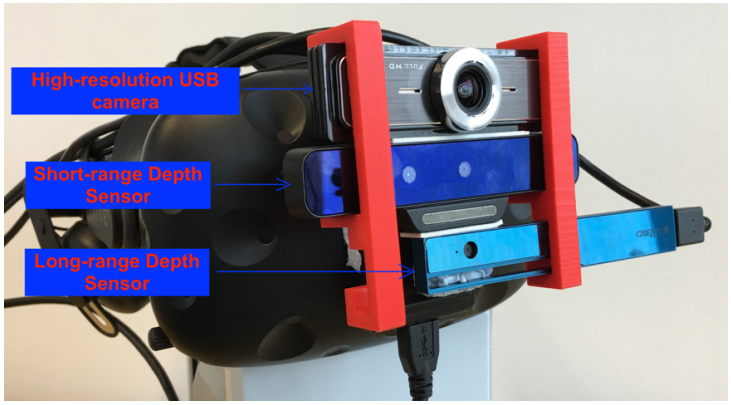


FIGURE 6.2: Local Headset setup

6.1.2 Tasks Design

Our study's basic idea was to ask the participants performing as the remote experts to guide the local worker to find target letters on LEGO models located at different locations around the local workspace. There were two separate workspaces in our study. The local task workspace was used for the local worker to perform physical tasks, and the remote expert workspace was for the expert to support remote guidance.

Figure 6.3 shows one image of the local task workspace. There were eight LEGO models randomly located in the workspace, along with some irrelevant objects to block the views of the LEGO models from each other. Each LEGO model had one unique color (either red, orange, light green, dark green, blue, yellow, white, or brown), so that objects could be searched for based on their colors. Each LEGO model had three labels with different colors and letters on it. The same as block colors, the label colors could be used for searching for the target label on the target object.

Before each trial started, the system randomly picked one model color among the eight possible colors as the target model color, and one label color on the target model. The

FIGURE 6.3: Scene of the local task workspace

model color and label color were then shown on the remote expert's VR display as the object that needed to be found. Following this, the remote expert needed to guide the local worker to find the target model first based on the model color, and then locate the target label on the model based on the label color. When the local worker finally saw the right label on the right model, the remote expert needed to read the four letters on the label and press the trigger on the controller to finish the trial. The system then automatically recorded the trial completion time and showed the next pair of target model and label colors to the remote expert. For each interface, the remote expert had to complete one training trial and five formal trials in total.

To guide the local worker, the remote expert first needed to find the target model himself/herself. The low-resolution 3D point cloud background provided an overview of the entire local workspace, which was a straightforward way for the remote expert to locate the model position. However, while searching for the label on the model, the resolution of the 3D background may not have been adequate. In this case, checking the real-time high-resolution foreground view would be the right choice, especially if the expert asked the worker to pick up and rotate the model for him/her to search.

From the study in Chapter 5, we found that experts' hand gestures would be limited by the depth sensor's field of view. The remote experts needed to stare at their hands in order to capture the guiding gestures, which could limit the expression of the guiding cues. During this study, instead of using hand gestures, the remote expert was required to hold one Vive controller in hand, which was rendered as a virtual wand in the VR world (Figure 6.1). By using the wand, the remote expert could point to an object to guide the local worker. Simultaneously, both the local worker and remote expert could

talk to each other for communication. For the experiment, we set both the local and remote users in the same large room, separated by a curtain, so that they could easily talk to one another. However, the remote expert was not allowed to directly describe the target model color or label color to the local worker. The local worker could also point to physical objects by using his/her hands within the view of the head-mounted sensor, which could be seen by the remote expert through the real-time foreground view.

6.1.3 Participants

Participants were recruited in pairs. During the study, one participant performed the role of a remote expert while the other participant performed the role of a local worker. After one participant experienced all the three interfaces, we asked the two participants to switch their roles to rerun the study. Since the local workers used exactly the same video see-through interface (introduced in Section 4.2) for all three conditions, we did not measure the local worker's performance in this user study, and only the feedback from the remote experts was recorded. Five pairs of people (ten feedback in total) took part in the pilot study, four women and six men, aged from 19 to 44. Most of them had previous experience with video conferencing systems, such as Skype, Snapchat, or WeChat, except one. All of the participants could identify the colors without any trouble.

6.1.4 Experiment Design

We used a within-subjects design, which required the participants to experience all the three interfaces. At the beginning of the user study, participants were given a general explanation about the features of the three interfaces and the procedures of the tasks in detail. After this, they were asked to wear the VR headset to start the study.

The participants were exposed to the three interfaces in random order. For each interface, participants had one training trial followed by five formal trials. For each formal trial, the system automatically recorded the trial completion time. After all the five formal trials of each interface were completed, participants were asked to answer the usability related interview questionnaire on a Likert scale from 1 (strongly disagree) to 7 (strongly agree). After the participants finished all the three interface trials, they were also asked to provide opinions about each interface's advantages and disadvantages and choose the interface they liked to use the most.

6.1.5 Experiment Results

Overall, 98.7% of the trials were completed correctly (only two errors of a total of 5*3*10=150 trials). We noticed that both errors were made by one participant who accidentally identified the orange LEGO model as a red model.

Figure 6.4 shows the average task completion time with standard error. A one-way repeated measures ANOVA was conducted to determine whether there was a statistically significant difference in the average task completion time among the three interfaces. There were no outliers, and the data was normally distributed for each interface, as assessed by the boxplot and Shapiro-Wilk test ($p > 0.05$), respectively. We found no significant difference in the task completion time, $F(1.280, 11.521) = 1.590$, $P = 0.231$, with average task completion time increasing from 44.26 ± 9.02 seconds while using FPV to 48.15 ± 14.02 seconds while using TPV and to 57.16 ± 27.88 seconds while using PCV.

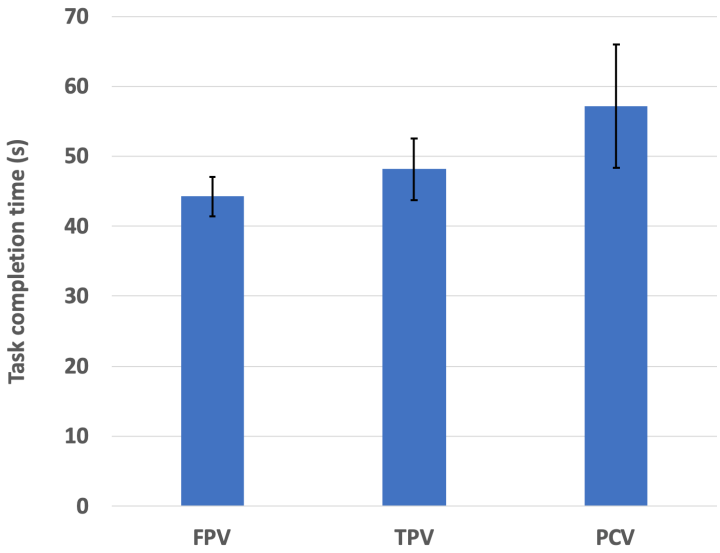


FIGURE 6.4: The average task completion time

Immediately after the trials for each interface, participants were asked to provide their subjective feedback on some usability rating questions. To compare the Likert scale ratings between the three interfaces, we used the Friedman test ($\alpha = 0.05$). For those results showing a significant difference between the three conditions, we ran post hoc tests for pairwise comparison using the Wilcoxon Signed-Rank test with Bonferroni correction applied ($\alpha = 0.0167$).

According to the Friedman test, there were significant differences in terms of "like to use" among the three interfaces ($\chi^2(2) = 6.414$, $p = 0.040$). However, the Wilcoxon Signed-Rank test indicated that there were no significant pairwise difference (TPV and FPV: Z =-0.276, p = 0.783; FPV and PCV: Z = -1.983, p = 0.047; TPV and PCV: Z = -2.209, p = 0.027).

The result also showed significant differences in terms of "easy to use" ($\chi^2(2)$ = 12.519, p = 0.002). Pairwise comparisons indicated that participants believed the PCV interface was significant hard to use than FPV interface (Z = -2.549, p = 0.011) and TPV interface (Z = -2.414, p = 0.016). No significant difference was found between FPV and TPV (Z = 0.000, p = 1.000) about which interface was easy to use. Furthermore, participants felt significant different physical stress while using the three interfaces ($\chi^2(2)$ = 13.867, p = 0.002). For pairwise comparisons, significant differences were found between FPV and TPV (Z = -2.555, p = 0.011), and between PCV and TPV (Z = -2.558, p = 0.011) on physical stress, but no significant difference between FPV and PCV (Z = -0.707, p = 0.48). For other usability rating questions, no statistically significant differences were found between the three interfaces (Helpful: $\chi^2(2)$ = 3.000, p = 0.223; Confident: $\chi^2(2)$ = 4.846, p = 0.089; Mental stress: $\chi^2(2)$ = 2.774, p = 0.250). Figure 6.5 shows the average rating for usability rating questions.

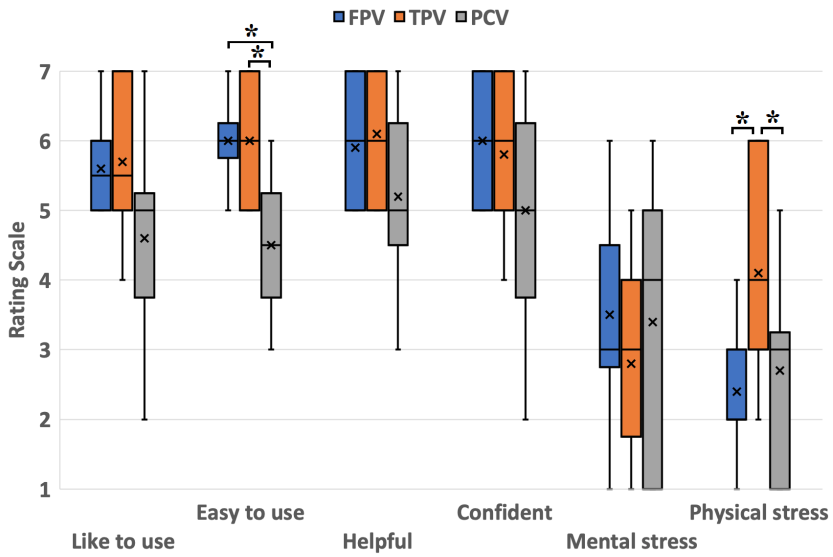


FIGURE 6.5: Results of the usability rating scale(∗: statistically significant difference)

In the post-experiment questionnaire (Figure 6.6), 50% of the participants chose the FPV interface as the one that they preferred most to use in a remote collaborative task. Statements by participants about their choices included: *"it was easy to check what the partner was working on"*, *"the other person's presence was obvious to me"* and *"this interface requires less physical movements in order to find the correct objects"*. Thirty percent of the participants selected the TPV interface as their first choice because: *"the view screen is bigger than the first-person view and image is more clear"* and *"this interface makes me feel the presence in the scenario"*. Only two participants out of ten said they preferred to use the PCV interfaces. They thought the PCV interface had some unique advantages, such as: *"the 3D point is direct and specific"* and *"it is very easy to find the target label in the first place"*.

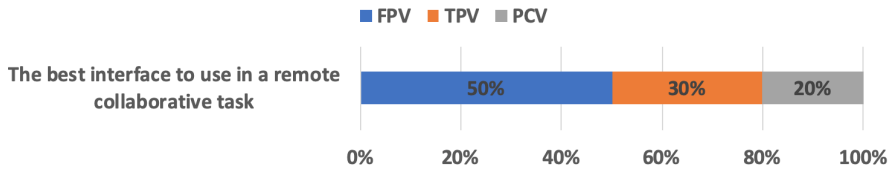


FIGURE 6.6: User preference rank for different types of interfaces

6.1.6 Discussion

To summarize the results, all three interfaces enabled the remote expert to guide the local worker on completing some physical tasks in the large workspace. However, no significant statistical difference was found in the average task completion time for each of the interfaces. Based on the subjective feedback, we believed that all three interfaces had their own advantages and disadvantages.

In terms of the FPV interface, participants provided positive feedback about seeing the 3D background and 2D foreground simultaneously. They thought it was straightforward for them to check the real-time feedback from the local worker through a 2D video window that always followed their view. They had the ability to monitor the local workspace changes through the duration of the tasks with less physical movement. However, some of them also mentioned that the first-person view window was small, and it was sometimes hard to see the view clearly.

For the TPV interface, as the 2D video window followed the local worker's viewpoint movement, participants indicated this interface was helpful for them to understand their partners' actions in the scene. It was easy for them to confirm whether or not their partner was following their commands. Furthermore, the 2D video window of this interface was more notable and apparent than the FPV interface. However, this interface required the remote experts to move a lot to catch the real-time local view, since they could only see it behind the local worker's virtual view frustum. Therefore, participants felt more physically stressful while using this interface than the other two views. Another issue reported by the participants was that this third-person view sometimes blocked the view of the 3D background in the VR space, and interrupted their guidance. As shown in Figure 6.7, when the local worker was too close to the workspace, the worker's view frustum, the 2D foreground video and the 3D background point cloud sometimes mixed together and blocked the views of each other.

PCV interface displayed the local changes precisely the same as where they took place in the local physical workspace. However, due to the narrow field of view of the depth

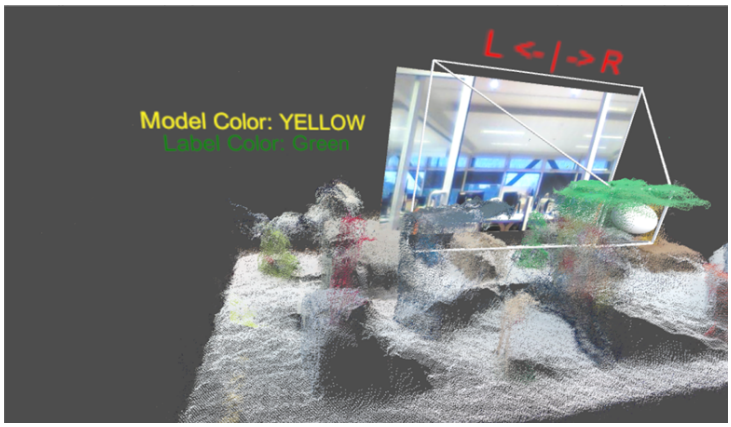


FIGURE 6.7: The local worker's view shown in the TPV interface. The local worker's view frustum blocked the view of the remote expert

sensor and the poor point cloud resolution (less than 320*340 points for each frame), it took more effort for the experts to guide the local workers to find the right target label. Participants turned to spent more time on completing the tasks while using this interface, but not significant. Therefore, they rated this interface as the most difficult to use during a remote collaborative process. Only two participants out of ten chose this interface as the one they most preferred to use. The other limitation reported by the participants was that they were not sure whether or not the local worker focused on the target model until the worker finally moved the model. While using the PCV interface, the experts could only know the worker's view direction based on the virtual view frustum, without one focus point. Furthermore, since the static background and changeable foreground point cloud were always mixed together, this may also decrease the usability of this interface design. Overall, this interface was considered significantly more challenging to use than the other two interfaces by participants.

6.2 Formal User Study

Due to the small sample size of our pilot study, we decided to re-run our user study to see if we could get a more significant statistic result. Since the resolution of the current real-time point cloud for the PCV interface was extremely low, and this point cloud was directly overlaid on top of the static background point cloud scene, it was hard for the experts to identify which was the background scene and which was the foreground view. Therefore, it was significantly difficult for experts to observe the local changes in the shared VR space compared to the other two interfaces. To deal with this problem, we designed an alternative interface to replace the PCV interface. Details of this new interface setup were described in the following section.

6.2.1 Experiment Setup

Due to the limitations of our PCV interface design, we changed this interface into one with a switchable view (SWV). As shown in Figure 6.8, while using this SWV interface, the user could switch his/her view between a static 3D point cloud reconstruction of the local workspace and a live 2D video of the local scene. Each of these two views covered the user's entire field of view at one time. During the tasks, the user needed to press one button on the controller to switch between these two views according to his/her own requirement. Since the USB camera used to capture the 2D live view was attached to the local worker's headset, we could only provide the local worker's first-person view as the 2D foreground. In this case, the remote expert experienced a viewpoint shift while switching between the 3D and 2D views.

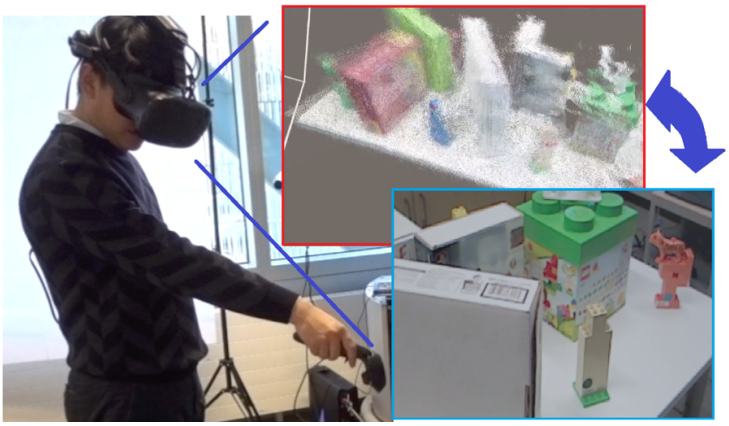


FIGURE 6.8: The switchable view interface

All of the three interfaces (FPV, TPV, and SWV) in our formal study combined a static 3D point cloud reconstruction to show the physical layout of the local workspace and a real-time 2D video to show the current changes of the objects in the local workspace. In this formal user study, we measured how well the 3D static background combined with a 2D live foreground could support remote collaborative tasks in one MR system setup.

We used the same task design as the pilot study. Participants performed as remote experts and guided the local worker to find the target label attached to specified LEGO models based on a pre-defined label color and model color. When the target label was located by the local worker, the remote expert needed to read the letters on the label. The detailed task process was discussed in Section 6.1.

We used a within-subjects design, which required the participants to experience all the three interfaces. Participants were recruited in pairs, one as the expert and one as the worker. At the beginning of the user study, participants were given a general explanation

about the features of all three interfaces and the procedures of the tasks in detail. Then they were required to put on the headset to start the study.

The participants experienced the three types of interfaces in random order. For each interface, participants had one training trial followed by five formal trials. After participants finished all the trials for one interface, they were asked to answer questions on the usability and social presence related questionnaire on a Likert scale from 1 (strongly disagree) to 7 (strongly agree). After the participants finished all the three interface trials, they were asked for their opinions about the advantages and disadvantages of each interface, and to choose one of the interfaces which they preferred to use most. Then, we asked the pair of participants to switch the roles and rerun the study.

Fifteen pairs of participants were recruited, seven women and twenty-three men, aged from 21 to 39. Most of them had previous experience with video conferencing systems, such as Skype, Snapchat, or WeChat, except one. Twenty-one people had experienced at least one AR or VR application. All of the participants could identify the colors without any trouble. Since each pair of the participants run the study twice, we received thirty feedback in total.

6.2.2 Experiment Results

Figure 6.9 shows the average task completion time with standard error. A one-way repeated measures ANOVA was conducted to determine whether there was a statistically significant difference in the average task completion time among the three interfaces. The data were normally distributed for each interface, as assessed by the Shapiro-Wilk test ($p > 0.05$). We found no significant difference in the task completion time, $F(2, 54) = 0.134$, $P = 0.875$, with an average task completion time of 29.98 ± 8.52 seconds while using FPV, 30.68 ± 11.73 seconds while using TPV and 29.36 ± 10.71 seconds while using SWV.

We also noticed that, compared to the pilot study (Figure 6.4), participants spent less time completing the guiding tasks while using FPV and TPV interfaces. The local workspace in the pilot study was smaller compared to the formal study; therefore, the objects we placed in the workspace seemed to be more close to each other, which could cause complicated occlusion issue than the formal study. This might be the reason why participants spent less time in the formal user study.

Participants were asked to immediately provide their subjective feedback on some usability rating questions after the five formal trials for each interface. A Friedman test

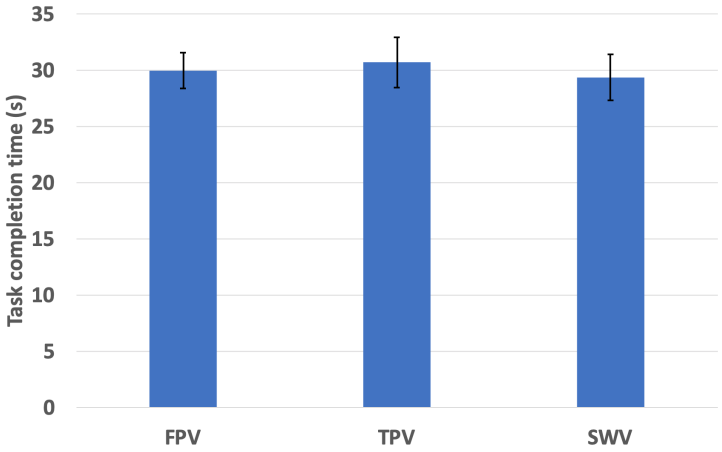


FIGURE 6.9: Average task completion time

(α = 0.05) was used to determine whether there was a statistically significant difference in the usability rating scales. For those results showing a significant difference between the three conditions, we ran post hoc tests for pairwise comparison using the Wilcoxon Signed-Rank test with Bonferroni correction applied (α = 0.0167).

According to the Friedman test, participants felt significant different mental stress while using different interfaces (FPV: median = 2.5, interquartile range = 1.000; TPV: median = 3.0, interquartile range = 2.250; SWV: median = 2.0, interquartile range = 1.000; $\chi^2(2)$ = 6.256, p = 0.044). Pairwise comparisons indicated that participants felt significant mentally stressed while using TPV interface than using SWV interface (Z =- 2.575, p = 0.01). No significant difference was found between TPV interface and FPV interface (Z =- 2.206, p = 0.027), and between SWV interface and FPV interface (Z =- 0.428, p = 0.668) in terms of mental stress. In addition, participants also experienced significant different physical stress while using different interfaces (FPV: median = 3.0, interquartile range = 1.250; TPV: median = 4.5, interquartile range = 2.250; SWV: median = 3.5, interquartile range = 3.000; $\chi^2(2)$ = 7.506, p = 0.023). For pairwise comparisons, significant differences were found between TPV and FPV (Z = -3.191, p = 0.001), and between SWV and FPV (Z = -2.576, p = 0.010) on physical stress, but no significant difference between SWV and TPV (Z = -0.492, p = 0.622). For other usability rating questions, no statistically significant differences were found among the three interfaces (Like to use: $\chi^2(2)$ = 0.194, p = 0.907; Easy to use: $\chi^2(2)$ = 0.506, p = 0.776; Helpful: $\chi^2(2)$ = 1.649, p = 0.439; Confident: $\chi^2(2)$ = 0.076, p = 0.963). Figure 6.10 shows the average rating for usability rating questions.

The social presence questionnaire (SoPQ) [48] we used in our study included four sub-factors: Co-presence (CP), Attentional Allocation (AA), Perceived Message Understanding (PMU) and Perceived Behavioral Interdependence (PBI). Each of these

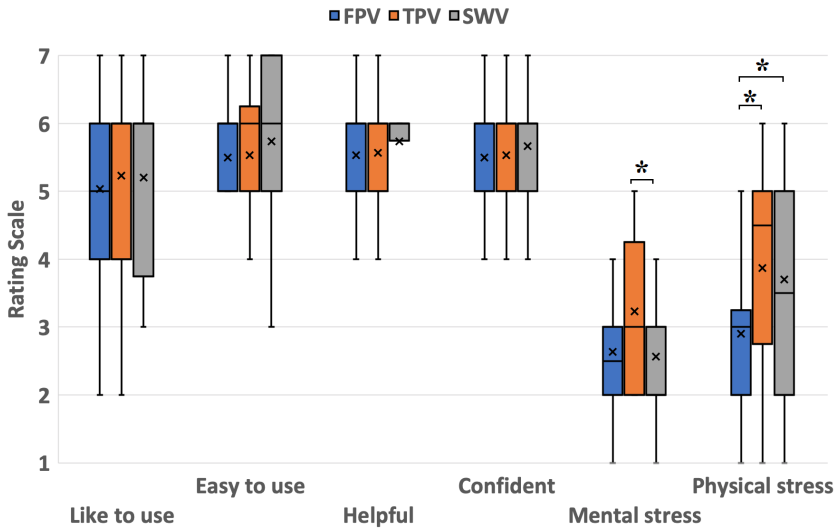


FIGURE 6.10: Results of the usability rating scale (∗: statistically significant difference)

sub-factors consisted of six closely interrelated questions scaled from 1 (strongly disagree) to 7 (strongly agree), which represented a group of Likert scale measurements. To analyze Likert scale data, we first calculated a composite score from the six questions for each sub-factor [110], then one-way repeated measures ANOVA was conducted to determine whether there was a statistically significant mean difference among the three view interfaces.

Overall, we found no significant difference in social presence over the three interfaces ($F(1.892, 54.861) = 0.635$, $p = 0.525$) as shown in Figure 6.11. We also analyzed each sub-factor and found statistically significant differences in CP over the three interfaces (FPV: mean = 5.433, standard deviation = 0.734; FPV: mean = 5.817, standard deviation = 0.666; FPV: mean = 5.400, standard deviation = 0.765; $F(2, 58) = 4.357$, $p = 0.017$). Post hoc analysis with a Bonferroni adjustment revealed that participants gained statistically significant better feeling in co-presence while using TPV compared to FPV (0.383(95%CI, 0.009 to 0.757), $p = 0.043$), and while using TPV compared to SWV (0.417(95%CI, 0.001 to 0.832), $p = 0.049$), but no significant difference between FPV and SWV (0.033(95%CI, -0.372 to 0.438), $p = 1.000$). For other sub-factors, we did not found any significant differences (AA: $F(1.445, 41.914) = 0.732$, $p = 0.445$; PMU: $F(2, 58) = 0.118$, $p = 0.889$; PBI: $F(2, 58) = 0.891$, $p = 0.416$).

After the study tasks, we also asked the participants to select the best interface they preferred to use for collaborative tasks. As shown in Figure 6.12, twelve participants chose the FPV interface as the one they most preferred to use, compared to nine participants who chose TPV, and nine participants who chose SWV. A chi-square goodness-of-fit test

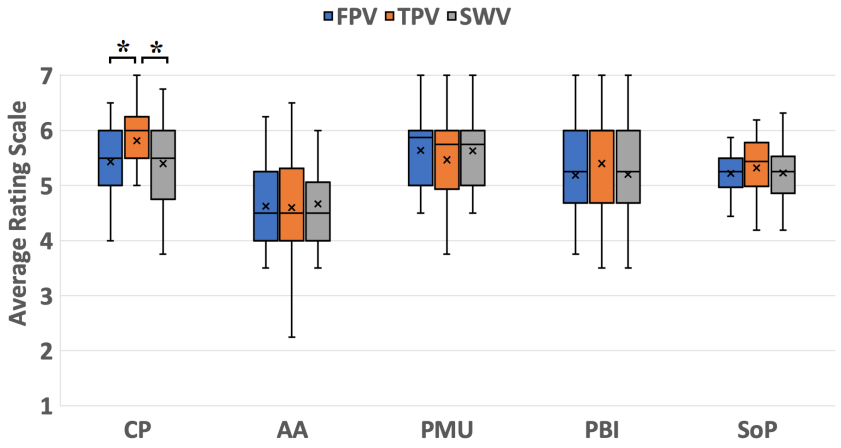


FIGURE 6.11: Results of the Social Presence questionnaire (CP: Co-presence, AA: Attentional Allocation, PMU: Perceived Message Understanding, PBI: Perceived Behavioral Interdependence, and SoP: Overall Social Presence; $*$: statistically significant difference)

was conducted on this user preference ranking. There were no statistically significant differences in the number of participants of user preference (($\chi^2(2) = 0.600$, p = 0.741).

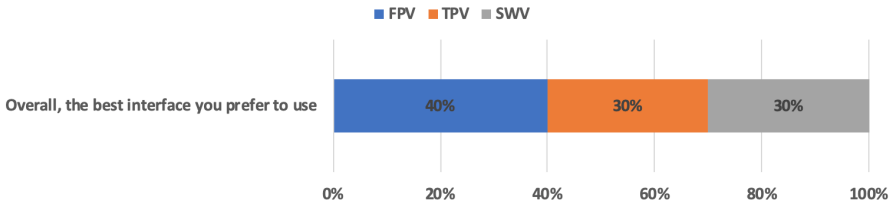


FIGURE 6.12: User preference about which interface they preferred best for collaborative tasks

6.2.3 Discussion

Overall, we did not find any significant differences in user performance based on the average task completion time. We need to further investigate the user performance by gathering other types of objective data in the following study. We believed that all three interfaces had their own advantages and disadvantages based on the subjective user rating questions and post-experiment questions.

Based on the result, we received almost the same feedback as the pilot study for the FPV interface. The FPV interface supported a continuous 3D static background view and 2D live foreground view at the same time. 40% of the participants ranked this view interface as the best interface to support remote collaboration. Compared to the other two interfaces, FPV showed a significant advantage in decreasing the experts' physical stress. The experts could observe both background and foreground view simultaneously through the entire collaborative process. Therefore, the participants could focus on the

guiding task. However, participants also pointed out that the live video window was small for clear visualization.

The TPV interface was useful for understanding remote partner's actions since it required the remote experts to locate the partner's view frustum in the shared VR world to catch the real-time feedback from the local side. In this case, the participants indicated that the local worker was significantly obvious to them and caught their attention during the task process while using this interface (Co-presence). However, the current local view attached to the worker's view frustum was quit unstable if the worker kept moving. In order to catch the local view, the remote experts were required to follow the worker's movement, which significantly increased the users' physical stress.

For the SWV interface, participants reported that it could provide a clear view of both the 3D point cloud scene and the 2D real-time feedback since each of these two views covered the entire visual field at one time. However, switching views also caused a viewpoint jump between the expert and the worker, which might interrupt the guiding process. Participants always spent some time on figuring out where the current viewpoint was after the view switching. We noticed that some of the participants would prefer to keep using the 2D live video view through the entire guiding process to avoid this viewpoint jump.

6.3 Conclusion

In this study, I presented a prototype remote guiding system with a hybrid interface combining a low-resolution 3D point cloud scene with a high-resolution real-time view. In this case, the remote expert had an overview of the local workspace with independent viewpoint control and the sense of spatial distribution. Meanwhile, they could also check the local worker's actions based on the real-time 2D video or 3D point cloud feedback.

Although we designed different interfaces to present this combination of the two types of views, there was no statistical difference in the task performance on average task time spent with our remote collaboration system. However, we found that the FPV interface required less physical movement, and the TPV interface could support better user experience in terms of co-presence, and the SWV interface had the advantage of reducing mental stress compared to TPV.

In the next study, I focused on analyzing the users' behaviors while using our MR

remote collaboration system with one hybrid interface. I intend to evaluate how remote users behave during different task stages. The answer to this question may help us to understand specific interface design requirements for different task purposes.

Chapter 7

User Behavior Analysis

In this research, I investigated how users behaved during the remote collaborative process while using an MR interface, especially for remote experts. Unlike the study from Chapter 6, view interfaces with different features were provided for the experts at the same time, so that they could decide which features to use based on the current task goal and situation.

In the first part of our user behavior analysis, we combined three viewing interfaces for the remote expert, 1) a 3D static point cloud view, 2) a 2D live first-person view, and 3) a 2D live God-like view. We measured the amount of time spent on each interface in a single task and collected usability feedback from participants. Based on the feedback, we found that the 2D live God-like view was limited by its FOV and was rarely used by users. Therefore, in the second part, we replaced the 2D live God-like view with a 360° panorama view and redesigned the experiment task to simulate an item arranging scenario in a large office room. We found that the remote experts prefer to learn the local physical layout and search for the targets with a global perspective from the 3D static view. The results also showed that the experts chose to use the 360° live view with independent view control rather than the 2D first-person view with high-resolution imagery to control the task procedures and check the local worker's actions. These two studies contribute to a more comprehensive understanding of interface design for MR remote collaboration systems in various guiding scenarios.

7.1 User Behavior Analysis: Part One

A remote collaboration task can typically be divided into several stages, including understanding the local physical layout, searching for the targets, guiding the local worker, and completing the task goal. Therefore, remote experts face different issues

during each stage of the task, and they may have different interface requirements to support them on specific stage goals.

In the user study from Chapter 6, we set up one switchable view interface (SWV) to compare with the other two interface designs (FPV and TPV). In the current user study, we further improved our interface design by integrating a God-like view into the SWV interface. In this case, we provided a combination of a group of three views (a 3D static point cloud background view, a 2D live first-person video view, and a 2D God-like view), and users could choose which view they needed to use based on the different situations they might face.

In this user study, we explored the basic principles of the interface design for different task stages while using an MR based remote collaboration system. We conducted a formal user study by analyzing remote experts' behaviors while using the combination of three different views during the collaborative process.

7.1.1 Experiment Setup

Interface Design

When remote experts work on collaborative guiding tasks, they often need to achieve a task goal following several steps. We describe these steps, together with the corresponding interface design requirements and solutions, in Table 7.1. We aimed to meet these interface requirements through the solutions we proposed using a switchable MR interface combined with three view modes.

In this study, we explored three views for the remote experts to understand the current local situation and provide real-time guidance to satisfy all the interface requirements from Table 7.1. Each of the views enabled the expert to observe the local workspace in one specific way:

- 3D point cloud view: The static 3D point cloud of the local scene was displayed in the remote expert's VR view to show the geometric layout of the local worker's workspace (see Figure 7.1 A). This view was designed to help learn the local geometry layout and search for target objects.

- 2D live first-person view: The local worker's first-person view was captured and streamed to the remote side in the form of 2D video (Figure 7.1 B). This view could enable the remote expert to understand the local worker's focal point with a high-resolution video view, which was helpful for detailed manipulation.

- 2D live God-like view: A top-down view of the entire local workspace was live captured and streamed to the remote side to give the remote expert a straightforward overview of the local work environment (Figure 7.1 C). This view was also designed to help search for target objects.

TABLE 7.1: Task process, interface requirement and design solution

	Task process	*Interface requirement*	*Design solution*
Step 1	Understand the task goal	Show the task goal to the remote expert	Display the task goal in the remote expert's VR view
Step 2	Learn the local physical layout	Show the view of the local workspace	Display the captured static scene of the local workspace in the form of a 3D point cloud or a God-like view
Step 3	Search for the target object or location	The same as above	The same as above
Step 4	Guide the local worker to the target object or location	Provide guiding tools	Pass the expert's pointing guidance and speech to the local side to enable communication
Step 5	Detailed searching or analyzing of the target object	Show focused view on the target object	Display the local worker's first-person view to the remote expert while the worker is looking at the target
Step 6	Complete the target goal, such as guide the local worker to finish the manipulation on the target object	Provide guiding tools	Pass the expert's pointing guidance and speech to the local side to enable communication

All three views were displayed as small foreground picture-in-picture windows on the right side of the remote expert's VR display (see Figure 7.1). During the tasks, the experts could freely switch between these views by clicking one button on a hand held controller. When one view was selected, it would automatically display as the background of the remote expert's view, and the corresponding foreground picture-in-picture window

would be highlighted with a red frame. In this case, the remote experts could make their own choices to use any of these views to complete the task goal based on the purpose of each task step.

FIGURE 7.1: The combination of three view interfaces. A: 3D point cloud view; B: 2D live first-person view; C: 2D live God-like view

Task Design

We set up the task process the same as the study introduced in Section 6.1 since this task design required the remote experts to follow all the six steps we defined in Table 7.1. To achieve the task goal, the remote experts needed to guide the local worker to find the right label on the right LEGO model in the local workspace.

As shown in Figure 7.2, we randomly put eight LEGO models in the local workspace together with some irrelevant objects to block the direct view of some of the LEGO models. Each LEGO model had one unique color (red, orange, light green, dark green,

blue, yellow, white, or brown) so that the remote expert could search for them based on their colors. In addition, each model was labeled with three different colors and letters at random positions. The remote expert could guide the local worker to the target model based on the model color and then find the right label based on the label color.

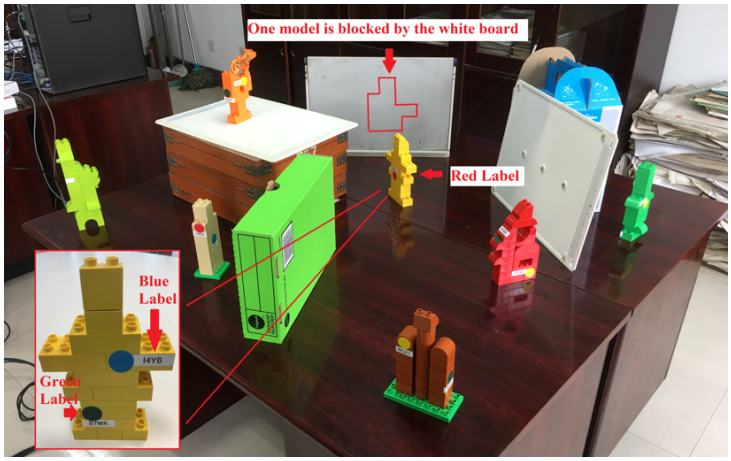


FIGURE 7.2: Scene of the local task workspace

At the beginning of each trial task, the system indicated the target object by randomly showing one model color among the eight model colors, and one label color among the three label colors in the remote expert's VR view (Table 7.1:step 1). Before the expert could start providing guidance, he usually needed to learn the spatial layout of the local workspace (Table 7.1:step 2) and find the target model by him or herself (Table 7.1:step 3). After this, the expert could guide the local worker to find and pick up the target model (Table 7.1:step 4). Based on the desired label color, the expert tried to find the target label on the model by asking the local worker to rotate the target model (Table 7.1:step 5). Finally, the expert asked the local worker to read the letters on the target label to finish the trial task (Table 7.1:step 6). To complete one user study, the remote expert had to finish one practice trial and five formal trials in total, in each of which they had to guide the worker to find one target model and read one target label.

During the tasks, the remote expert was required to hold a VIVE controller in one hand, which was rendered as a virtual wand in the VR world. Using the wand, the remote expert could point to an object to guide the local worker. We used a controller instead of natural hand tracking here because hand tracking was not accurate enough and could only be tracked over a small range (also check Chapter 5). Both the local worker and remote expert could talk to each other. For the experiment, we set both the local and remote users in the same large lab room, separated by a curtain. However, the remote expert was not allowed to directly describe the target model color or label color to the local worker. The local worker could also point to physical objects by using his/her

hands within the view of the head-mounted sensor, which could be seen by the remote expert through the first-person view interface. By pressing the trigger on the controller, the remote expert could freely switch between the three interface views.

Participants

The participants were recruited in pairs: one for the role of the expert and the other for the role of the worker, and then exchanged the roles. Since the local worker used the same video see-through interface no matter which view was active on the remote side, we did not measure the local worker's performance in this user study. The study results were only recorded when the participants played the role of remote expert. A total of thirty-eight people took part in the study, twelve women and twenty-six men, aged from 20 to 43. Most of them had previous experience with video conferencing systems, such as Skype, Snapchat, or WeChat, except for two people. However, fifteen of them had not used any VR or AR applications before. For those who had VR or AR experience, only two of them used these kinds of applications daily, while others had only tried them several times. All of the participants could identify the object colors without any trouble.

Data Collection

Task completion time was collected for the researchers to analyze user performance. In addition, the duration of each interface's use during one trial and the interface switching frequency were also recorded for the objective analysis of user behavior.

After the participants finished all five formal trials, they were asked to answer questions in a questionnaire on a Likert scale from 1 (strongly disagree) to 7 (strongly agree). Some of the questions were related to usability performance, such as "I think this system was easy to use", "I felt confident using the system", and "I felt the interface was mentally stressful". We also asked the participants to choose their preferred view interface under various categories related to the different task purposes covered in Table 7.1.

7.1.2 Experiment Results

The collected data consisted of both subjective feedback and objective measurements from the thirty-eight participants. We analyzed interface-related behaviors when a remote expert provided guidance on the local physical tasks, and we had a further discussion to address some other observations.

We assumed the participant performing as an expert always knew what to do and what was correct during the tasks, which simulated a real-world situation. In a real guiding task, the expert's role was to lead the worker to achieve the task goal in the right way.

In this case, the outcome of the task was always considered to be correctly completed. Therefore, we did not analyze the task accuracy in this user study. If an expert confirmed that the local worker read the correct letters on the right model, no matter whether the model and label colors were the same as those provided by the system, we considered that the task had been correctly completed. In fact, we noticed that only one participant accidentally identified the orange LEGO model as a brown model.

The results showed that participants spent, on average, 52.73 seconds (sd = 22.38) to complete one trial task. We also measured the average duration of use for each type of view during one trial task. From Figure 7.3, we see that participants spent almost the same time with the 3D point cloud view (an average of 25.19 seconds, sd = 9.94, 48%) and the 2D first-person view (an average of 25.07 seconds, sd = 19.09, 47%). However, the 2D God-like view was rarely used (an average of only 2.46 seconds, sd = 2.46, 5%). With a one-way repeated measures ANOVA, the average duration of use for the three views was statistically significantly different ($F(1.298, 48.041) = 43.065$, $p < 0.0005$). For pairwise comparisons, there was a significant difference in the average duration of use between 3D point cloud view and 2D God-like view ($p < 0.0005$), and between 2D first-person view and 2D God-like view ($p < 0.0005$), but no significant difference between 3D point cloud view and 2D first-person view ($p = 1.000$).

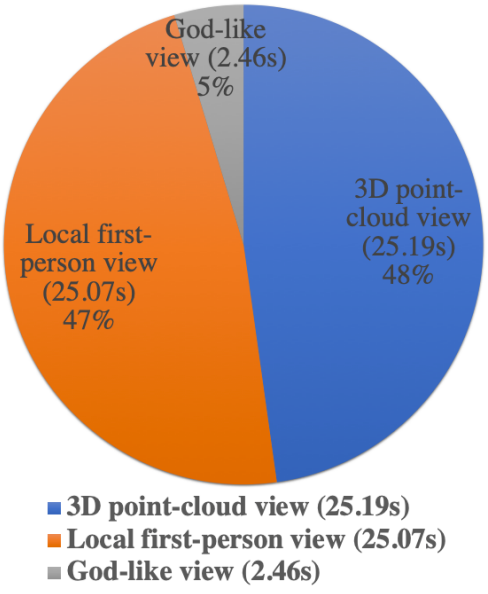


FIGURE 7.3: Average duration of use of each view interface

In our study, participants switched views among different view modes about 5.51 times (sd = 4.07) per trial on average. Most of the participants (31 out of 38) switched views during the tasks in order to take advantage of each view type for completing the goals of different task steps. However, five participants chose to use only the 3D point cloud

view during the entire task process, while two participants chose to use only the local 2D first-person view (Table 7.2). Those participants that only used the point cloud view spent less time (mean = 32.2 seconds) on completing the tasks. On the other hand, only using the local first-person view (mean = 67.6 seconds) took much more time than the other two interfaces.

TABLE 7.2: Interface choices of users

	Number	*Percentage*	*Avg. Task Time(s)*
Switching interfaces	31	81.6%	55.1
Using point cloud view only	5	13.2%	32.2
Using first-person view only	2	5.3%	67.6

Figure 7.4 showed the rating feedback for the usability test on a 7-point scale (1: strongly disagree – 7: strongly agree). The results showed most of the participants rated positively on Q1 to Q6 (Q1: median = 5, mode = 6; Q2: median = 6, mode = 6; Q3: median = 6, mode = 6; Q4: median = 6, mode = 6; Q5: median = 6, mode = 6; Q6: median = 6, mode = 6). One-sample Wilcoxon Signed-Rank test showed participants rated significant higher than neutral level rating for those questions (Q1: $Z = 3.414$, $P = 0.001$; Q2: $Z = 4.645$, $P < 0.05$; Q3: $Z = 5.344$, $P < 0.05$; Q4: $Z = 4.844$, $P < 0.05$; Q5: $Z = 5.308$, $P < 0.05$; Q6: $Z = 5.309$, $P < 0.05$). For the questions on mental and physical stress, results showed the remote experts' feeling as Q7 (median = 2.5, mode = 2) and Q8 (median = 3.5, mode = 2). One-sample Wilcoxon Signed-Rank test showed participants rated significant lower than neutral level rating for Q7 ($Z = -2.871$, $P = 0.004$), but no significant difference was found for Q8 ($Z = -1.515$, $P = 0.13$).

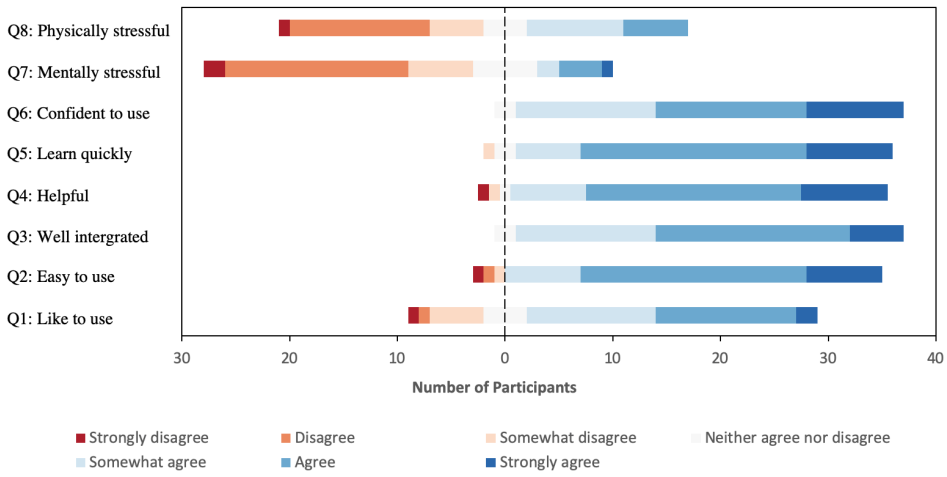


FIGURE 7.4: The usability rating feedback

In addition, we also asked the participants to choose one of the three views we provided as the best approach for remote collaboration according to different task objectives

(Figure 7.5). These task objectives covered the task steps we mentioned in Table 7.1. For example, item searching happened during step 3 and step 5, guiding happened during step 4 and step 6, detailed manipulation happened during step 6, and communication happened through all the six steps. A chi-square goodness-of-fit test was conducted to determine whether an equal number of participants chose each of the three view interfaces as the best based on different task objectives. The minimum expected frequency was 12.7. The results indicated that the three view interfaces were not equally preferred by the participants for all types of task objectives (target searching: $\chi^2(2) = 8.579$, p = 0.014; communication: $\chi^2(2) = 19.632$, $p < 0.05$; guiding: $\chi^2(2) = 8.895$, p = 0.012; detailed manipulation: $\chi^2(2) = 38.105$, $p < 0.05$).

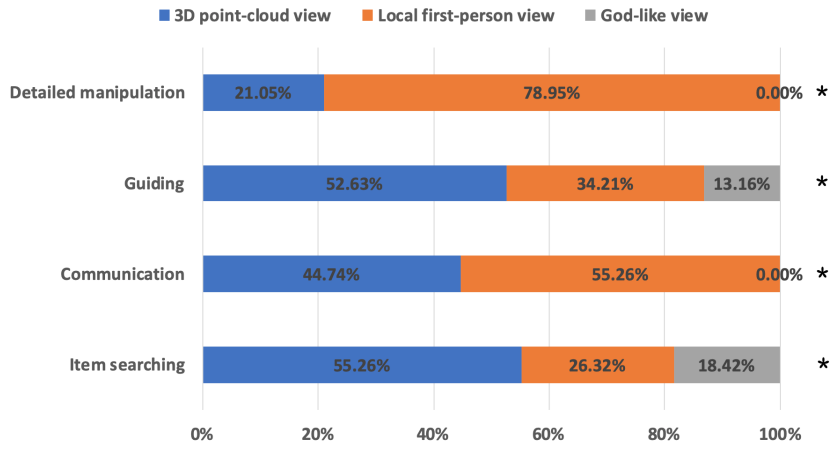


FIGURE 7.5: User rank for different types of tasks (∗: statistically significant)

Over half of the participants (55%) chose the 3D point cloud view as the most useful one for target searching (Local 2D first-person view: 26% and 2D God-like view: 19%). For communication, 55% of the participants thought that the local 2D first-person view was more natural, while 45% chose the 3D point cloud view, and none of them chose the 2D God-like view. In terms of guiding, the 3D point cloud view received the highest rank (53%) in contrast to the local 2D first-person view (34%) and the 2D God-like view (13%). At the same time, most participants (79%) selected the local 2D first-person view as the most useful for detailed manipulation. In particular, none of the participants thought that the 2D God-like view was helpful for detailed manipulation.

7.1.3 Discussion

Based on our research results, we found that our remote-side participants preferred to view the 3D point cloud replica of the local space rather than the 2D God-like view while trying to understand the spatial layout of the local working environment and searching for the targets (Figure 7.5: item searching). This could be because the 3D point cloud

view provided participants with a more natural way to interact with objects and perceive spatial distribution, similar to how they would do in the real world. For example, to check the surrounding situation, one participant usually first located himself/herself in the environment as a reference, and the locations of other objects were considered relative to this reference. Participants even said, *"I can feel and touch the objects on the local side."* In fact, there was no tactile sense, but they felt this feeling while immersing themselves in the shared 3D point cloud VR scene. Moreover, the experts could have a fully independent viewpoint from the local worker, which was a critical approach for increasing the remote collaborative experience, as discussed in Chapter 2. However, the 3D point cloud scene was static and was captured before the trial tasks started, and with a low resolution. While the users performed detailed manipulation of the target objects, such as finding and reading letters on the LEGO models, this view was no longer useful once the models had been moved. In this case the experts required a clear real-time view of the target objects.

The 2D God-like view was designed to be helpful for target searching in one collaborative task. However, based on the average duration of use of each view interface (Figure 7.3) and user preference ranking (Figure 7.5), this view was rarely used during the study. From the post-experiment feedback of the question "What issues are there for each interface?", we could see that the God-like view had several problems: (1) It was captured by a single fixed-view camera with a narrow field of view. Although it could provide a useful overview, some objects were still wholly or partly blocked by others (as shown in Figure 7.2). (2) In order to capture the entire view of the room-scale workspace, the camera needed to be placed high enough; therefore, small objects might have been too hard to identify. (3) It was hard for the experts to navigate the local worker to the target location since they might lose their sense of direction while switching to the 2D God-like view. To overcome this problem, participants suggested that we should set up multiple God-like cameras from different angles or one 360° camera to capture the local workspace, which might increase the usability of this view interface.

In our test, most participants chose to use the local 2D first-person view to read the letters on the target LEGO models (Figure 7.5: detailed manipulation). This view interface supported a high-resolution image of what the local worker was looking at; however, some participants also indicated that the local first-person view was quite unstable, and the field of view was quite narrow. For example, one participant, *"without the image-stabilization system, I can quickly feel dizzy while using the 2D first-person view."*

Figure 7.1 shows the view design of our system. In order to show the experts what

view options the system supported, all three views were always shown on the right side of the experts' VR display as picture-in-picture windows, no matter which view was active. From the feedback of the five participants who only used the 3D point cloud view to complete the tasks (Table 7.2), we found out that in fact they used the active 3D point cloud view for target searching and the non-active local 2D first-person view from the small picture-in-picture window for label searching and letter reading. A previous study [71] showed that remote experts expressed a strong preference to the view which provided the greatest sense of control. In our study, view switching provided participants with a sense of control. However, it may also increase the participants' cognitive load in choosing which view interface to use for the current task purpose. Therefore, participants spent less time on completing the tasks while using a 3D point cloud view only than switching views since they did not waste time on choosing which view to use.

The two participants who used the local 2D first-person view only thought that the low-resolution point cloud was not capable of supporting them in target searching. They would like to have had a more clear local view throughout the whole task process. However, the local 2D first-person view was captured by one head-mounted sensor, and the field of view was narrow. The expert did not have an independent view control compared to the other two interfaces. Prior work [111] showed that increased view independence led to faster task completion time and a decrease in the amount of time spent on communicating. Therefore, because of the limited view independence of the 2D first-person view, the participants spent more time on communication as well as the overall task completion time in our study.

In addition, we did not measure the local workers' performance as they used the same video see-through interface to follow the remote guidance and interact with the surrounding environment. However, we still noticed that the local workers always moved carefully and slowly to ensure they would not hit the objects in the physical world. They reported that *"the position of the object in the real world was not the same as I saw through the VR display."* The view provided by the video see-through display was narrow and hard for them to perceive the objects' depth. In other words, the video see-through display separated the local workers from their surrounding physical environment and led to incorrect spatial perception.

7.2 User Behavior Analysis: Part Two

In this part, we first tried to improve our system design to address the limitations found in the study from the first part, such as using AR glasses for the local worker, replacing the 2D God-like view with a 360° panorama view, and supporting world stabilized annotation for guiding cues. We then conducted our user study based on a device organizing task in a large office space, which simulated a real collaborative scenario where experienced staff guided new staff to organize a set of new devices just arrived in the office.

Since our system supported a fully independent sense of control for the remote experts, we intended to check if the experts' behaviors followed each of our view interfaces' design goals. We have the following hypotheses related to each task process we described in Table 7.1:

H1 : The experts will prefer to use a global view to learn the local physical layout (Step 2) based on usability evaluation and user experience analysis;

H2 : The experts will prefer to use a global view to search for the targets (Step 3) based on usability evaluation and user experience analysis;

H3 : The experts will prefer to use a global view to guide the local worker (Step 4) based on usability evaluation and user experience analysis;

H4 : The experts will prefer to use a real-time high-resolution view to check the local situation in a focused area (Step 5) based on usability evaluation and user experience analysis;

7.2.1 Experiment Setup

Based on the results and user feedback from the previous user studies, we discovered some system design limitations of our original MR based remote collaboration system, as shown in Table 7.3. Due to these limitations, the capabilities of our system might not meet the performance and usability requirements. For example, the workers require better spatial perception to interact with physical objects, and the experts need a clear 3D view to identify the targets. Therefore, the user's behavior sometimes was different from what we expected. In this section, we introduce modifications to our original remote collaboration system design, which could further improve the user experience of one remote collaborative task.

TABLE 7.3: MR based remote collaboration system design limitations

Hardware:	
Video see-through VR headset	1) Narrow field of view;
	2) Separate the local worker from the physical world;
View interface:	
3D point cloud view	1) Static;
	2) Low resolution;
2D first-person view	1) Narrow field of view;
	2) Unstable;
2D God-like view	1) Narrow field of view;
	2) Hard to navigate the local worker;
Virtual cues:	
Guiding cues	1) Only support pointing with one virtual wand;
Other cues	1) View frustum always block the 3D point cloud background

Local system setup

Instead of wearing the HTC Vive VR headset, the local worker wore AR glasses (Magic Leap 1 [1]) as shown in Figure 7.6. The Magic Leap display provides an optical see-through view for the local worker. Therefore, the worker could directly see and interact with their surrounding physical environment with virtual remote guidance cues overlaid on top of the real world. We also rendered the remote expert's head as a red avatar with a semi-transparent view frustum to show the expert's location and view orientation. In this case, the worker could feel like he or she was sharing the same workspace as the remote expert.

Remote system setup

Our updated system design still focused on using three types of views (Figure 7.7) to support the remote expert in learning the local situation and providing remote guidance via the VR display. Each of the views was designed to evaluate some of the hypotheses we described above:

- 3D static view (Figure 7.7 A): The local workspace was captured and displayed as a combination of the 3D static point cloud and pre-defined 3D mesh models. This view was designed to help the remote expert learn the local physical layout (**H1**)

[1] https://www.magicleap.com/

FIGURE 7.6: The local system setup. A: Local worker's optical see-through view. Remote guidance cues overlaid directly on top of the real world; B: The local worker directly sees and interacts with the local physical workspace; C: The remote expert's head avatar and semi-transparent view frustum

and search for the targets (**H2**). The remote expert may prefer to guide the worker by using this view (**H3**).

- 2D live first-person view (Figure 7.7 B): The local worker's first-person view was captured by the Magic Leap camera and streamed to the remote side in the form of live 2D video. This could enable the remote expert to understand the local worker's focal point with a high-resolution video view, which helped with detailed manipulation and the task process management (**H4**).

- 360° live God-like view (Figure 7.7 C): The local workspace was captured and streamed to the remote side in the form of a 360° live video. The view was also considered to help the remote expert learn the local physical layout (**H1**) and search for the targets (**H2**).

It was challenging to use one depth sensor to capture detailed information of one large workspace due to hardware and software limitations, such as low resolution, failure of depth capture, and failure of data stitching. In our study, we combined the 3D point cloud (objects on tables) and some pre-defined 3D mesh models (tables and walls) to present the local environment in the VR space, which could enhance spatial immersion for the remote expert (Figure 7.8). We also updated the God-like view from a limited 2D view to 360° panorama view (captured by the RICOH THETA V [2] and streamed in 4K resolution), and aligned this 360° view to the 3D static view. Therefore, the remote

[2]https://theta360.com/en/about/theta/v.html

experts could smoothly switch between these two views without losing their sense of direction.

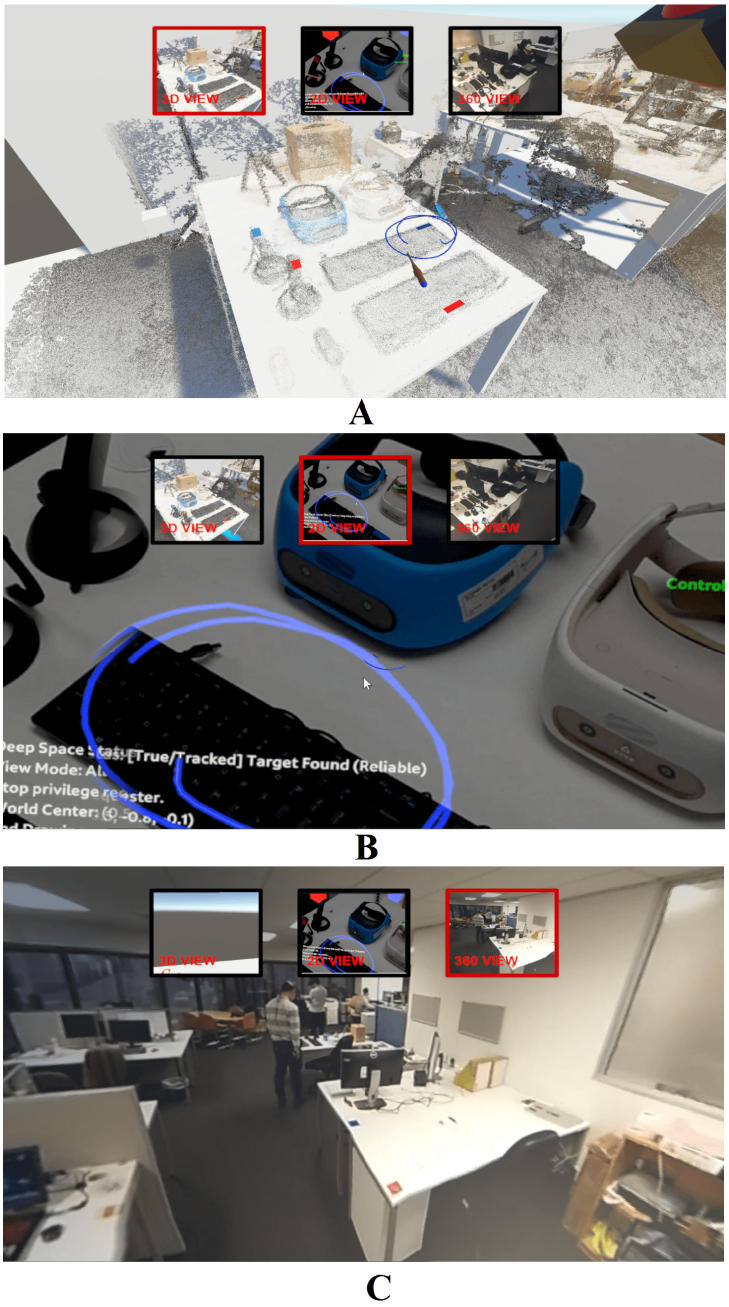


FIGURE 7.7: The combination of three view interfaces. A: 3D static view; B: 2D live first-person view; C: 360° live God-like view

We also enhanced the guiding cues by enabling the remote experts to draw lines in the 3D world. By pressing the trigger button of the controller holding in hand and moving the controller (rendered as a virtual wand in the VR world) in space, the experts could draw lines in the 3D VR space. These line annotations were world stabilized and could be observed by the local worker through the AR glasses (Figure 7.6). In this case, during

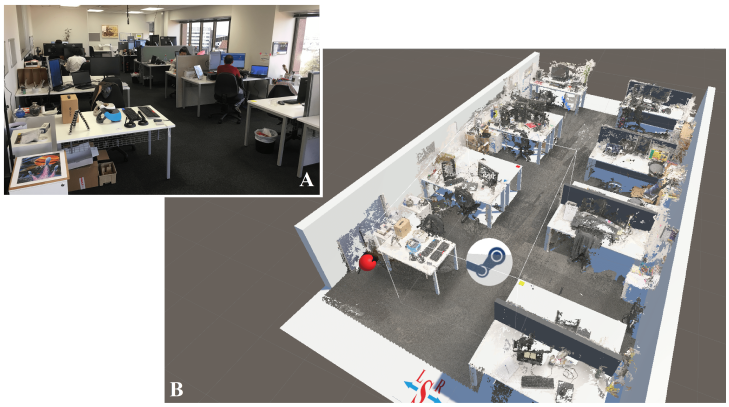


FIGURE 7.8: Local workspace. A: Local physical workspace; B: 3D static reconstruction of the local workspace

the collaborative task process, besides pointing and speech, the remote expert could also use annotations to indicate the target object or navigate the worker to the target location. Furthermore, we rendered the partner's virtual head avatar identical to the worker's head location and orientation in the physical world (Figure 7.9). Therefore, both the worker and the expert were brought into the same VR environment, having a similar collaborative experience as if they were face-to-face.

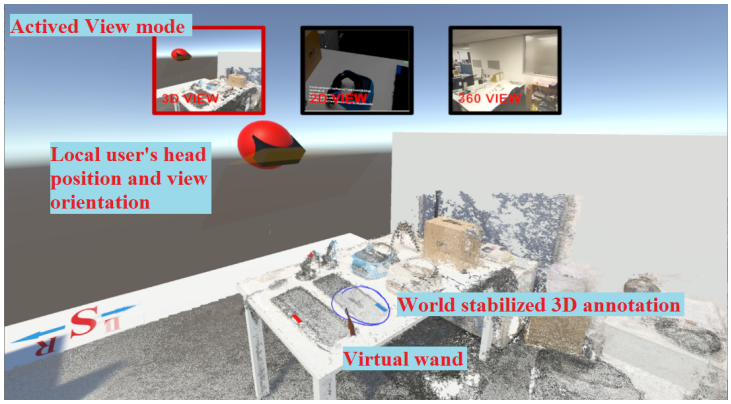


FIGURE 7.9: The remote expert's VR view

Task design and study environment

The user study was conducted in two separate spaces. The local worker was in an office room sized approximately 12m x 6m (Figure 7.8 A), while the remote expert was in a space beside the office room with an approximate size of 3m x 3m. The office room that the local worker stayed was the main task space for collaboration, including six tables along with the wall (left side) and seven tables along with the windows (right side). The local worker could freely move inside this room. The 3D reconstruction of this room was generated as a combination of 3D point cloud and mesh models before the study. We set up one 360° camera on the ceiling, which enabled 360° overviews of the entire office room. The remote expert's workspace was mainly a free walking space

with no furniture, which prevented the expert from hurting himself while wearing the VR headset. However, the size of the remote workspace was much smaller than the local workspace. In order to ensure the remote expert could reach any position in the reconstructed local virtual scene, we supported a teleport function (Figure 7.10). When the expert pointed the wand to the ground, he/she could see a blue circle and teleport to that position with a button pressed on the handheld controller.

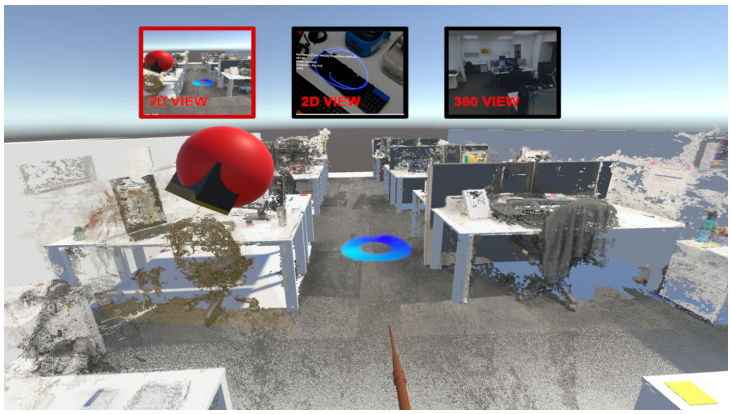


FIGURE 7.10: Teleportation in the VR space

In this study, we simulated a simple real work scenario. We assume that some new devices arrived at the office, and a worker was asked to organize and arrange them in the office. However, the worker did not know which tables the devices should be put on. Therefore, a remote expert would provide help with organizing these devices. This trial task could also cover the task process step we defined in Table 7.1. Initially, we put all the devices (target objects) on one table in the office (Figure 7.11 A). The expert was given a description of the target device and the target location where the device should be placed (step 1). Then he needed to first find the target device on the table (step 2 & 3), and instruct the worker to grab it (step 4). After this, the expert should search for the target location (step 2 & 3), and navigate the worker to this location (step 4). Finally, when the worker reached the target location, the expert asked the worker to place the device at the target location and checked if the worker put the device correctly (step 6). The trial task ended once the remote expert confirmed that the target device was placed in the correct required location. For this task design, we did not require the remote expert to manipulate or analyze the target object; therefore, we ignored step 5 in Table 7.1.

There were two task conditions, depending on how the target location was described. The first type of task (Visual Condition) instructed the expert to find the target location based on the visual description, such as color (Figure 7.11). For example, "Please put the blue headset in the place with a yellow marker." Since the 3D reconstruction was

aligned to the real-world coordinate system, the expert could use either of the three view interfaces to search for the target location. The second type of task (Spatial Condition) described the spatial configuration and relationship of the target location. For example, "Please move the white headset to the center of the third table from the left." This was done to determine if the task description also played a role in affecting the remote collaboration performance.

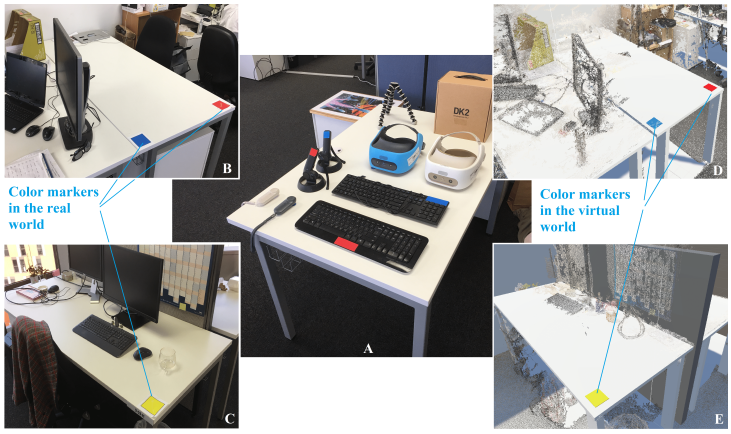


FIGURE 7.11: Task setting. A: Target devices were placed on one table and needed to be arranged in the office room. B & C: The color markers in the real world were used to indicate the target locations based on visual description. D & E: The color markers in the virtual world were used to indicate the target locations based on visual description

The task description was shown in the remote expert's VR view as text. To complete one study, the remote expert had to finish one practice trial and three formal trials for each task condition. In each trial, they had to guide the worker to grab the correct device and then place it in the right target location. The order of task conditions was counterbalanced to reduce bias.

Data collection and measurements

We collected both objective and subjective feedback from each task condition. Since the local workers used the same AR interface throughout the whole study, we did not measure the local workers' performance in this user study. The study results were only recorded when the participants played the role of the remote expert.

We assumed that the participants acting as the remote expert always knew what to do and what was correct during the tasks, which simulated a real-world situation. In this case, the outcome of the task was always considered to be correctly completed. Therefore, we did not measure the task accuracy in this user study. We collected the task completion time as the indication of task performance, and the duration of use of each view interface and the view switching frequency as the objective measure for

user behavior. We also tracked the experts' movement while using each of the view interfaces.

We used questionnaire responses to collect subjective feedback from participants. For each task condition, we measured user experience through a Single Ease Question (SEQ) [103], and some custom rating items from Theophilus et al. [113]. To evaluate the usability of our MR based remote collaboration system, we used the System Usability Scale (SUS) [13]. We also used the Networked Minds Social Presence Questionnaire (SoPQ) [48] and the MEC Spatial Presence Questionnaire (SpPQ) [120] to measure the sense of being together, and the Simulator Sickness Questionnaire (SSQ) [59] to measure motion sickness. At the end of the questionnaire, participants were asked to rank the three views under various categories and explain why they made these ranking choices.

Study procedure

Before the study started, the researcher briefly introduced the study information to the participants and asked them to sign the consent form once they decided to participate. A pair of participants were recruited together, one performed as the expert, and the other performed as the worker. The expert was asked to complete a pre-experiment questionnaire with general information and an SSQ to measure the ground level of motion sickness. Then a training session was held for the participants to learn how to switch the view, draw annotations and teleport in the space, which took around five minutes.

During the task process, the worker was asked to wear the Magic Leap glasses and interact with the physical world following the expert's guidance. The expert was asked to wear the HTC Vive HMD with the option to switch between three view interfaces at any stage of the task process. The expert could use pointing, speech, and annotation to communicate with and guide the local worker on completing the tasks. Once they completed one task condition, including one practice trial and three formal trails, the expert answered a per-condition questionnaire (SEQ questionnaire). Then, they proceeded into the second task condition. After completing this task condition, the expert was asked to fill another per-condition questionnaire followed by SSQ questionnaire and a post-experiment questionnaire. Since the expert used the same VR interface for both task conditions, the results for social presence, spatial presence, and system usability were not affected by this independent variable. In this case, we only asked the remote expert to answer the SoPQ, SpPQ, and SUS questionnaires in the post-experiment questionnaire.

We also asked the worker to provide some subjective feedback by interviewing him/her

after the study. This interview was audio recorded. Then, we asked the pair of participants to switch the roles and rerun the study.

7.2.2 Experiment Results

A total of 28 people took part in the study, 11 women and 17 men, aged from 15 to 35. Most of them had previous experience with video conferencing systems, such as Skype, Snapchat, or WeChat, except for three people. Five of them had not used any VR or AR applications before.

Task performance

A two-way repeated measures ANOVA was run to determine the effect of different task conditions over view interfaces on the average task completion time. Analysis of the data showed that there was normality, as assessed by the Shapiro-Wilk test of normality. There was a statistically significant two-way interaction between view interfaces and task conditions, $F(2, 54) = 5.881$, $p = 0.005$. Therefore, we run post hoc analysis the main effects of View interfaces and task conditions.

A one-way repeated measures ANOVA was used to determine whether there was a statistically significant difference in average task completion time between Visual Condition and Spatial Condition. The results showed that, overall, participants spent more time with Spatial Condition (66.36 ± 20.96 seconds), as opposed to Visual Condition (59.03 ± 18.192 seconds), a statistically significant increase of 9.33 (95% CI, 0.09 to 18.58) seconds, $F(1,27) = 4.289$, $p = 0.048$ (Figure 7.12). We also evaluated the average task time spent on each interface between these two task conditions. There was a significant difference of the time spent in the 3D static view ($F(1,27) = 7.687$, $p = 0.010$). No significant difference was found for 2D first-person view ($F(1,27) = 0.431$, $p = 0.517$) and 360° God-like view ($F(1,27) = 1.087$, $p = 0.306$).

Since our system enabled the remote experts to switch between three view interfaces at any stage of the task process, participants might behave differently based on their personal preferences. Time spent on each view interface played an essential role in showing the participants' focuses and interests. We used a one-way repeated measures ANOVA to determine whether there were statistically significant differences in average task time over three view interfaces during one task process. Data are mean $\pm$ standard deviation unless otherwise stated.

While participants searching for the target location based on the visual description, the average duration of use showed a statistically significant difference in our device

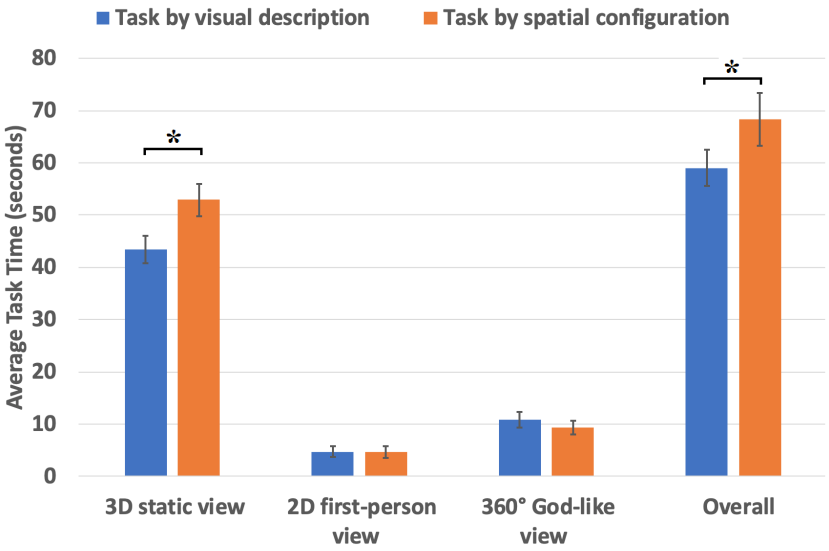


FIGURE 7.12: Average task completion time for each task condition (∗: statistically significant difference)

organizing task over the three view interfaces (F(1.497, 40.414) = 154.82, p < 0.0005), with 43.42 ± 13.34 seconds in the 3D static view, 4.77 ± 5.15 seconds in the 2D first-person view, and 10.83 ± 7.70 seconds in the 360° God-like view (Figure 7.13). Post hoc analysis with a Bonferroni adjustment revealed that participants statistically spent significantly more time in the 3D static view than the 2D first-person view (38.65 (95% CI, 32.19 to 45.11) seconds, p < 0.0005), and more time in the 3D static view than the 360° God-like view (32.60 (97% CI, 25.43 to 39.76) seconds, p < 0.0005), and more time in the 360° God-like view than the 2D first-person view (6.06 (95% CI, 2.05 to 10.06) seconds, p = 0.002).

While participants searching for the target location based on the spatial configuration, the average duration of use showed a statistically significant difference in our device organizing task over the three view interfaces (F(1.383, 37.343) = 28.81, p < 0.0005), with 52.90 ± 16.10 seconds in the 3D static view, 4.26 ± 4.94 seconds in the 2D first-person view, and 9.19 ± 6.18 seconds in the 360° God-like view (Figure 7.13). Post hoc analysis with a Bonferroni adjustment revealed that participants statistically spent significantly more time in the 3D static view than the 2D first-person view (48.65 (95% CI, 40.90 to 56.40) seconds, p < 0.0005), and more time in the 3D static view than the 360° God-like view (43.72 (95% CI, 36.79 to 50.64) seconds, p < 0.0005), and more time in the 360° God-like view than the 2D first-person view (4.93 (95% CI, 1.13 to 8.74) seconds, p = 0.008).

Overall, participants spent, on average, 48.17 seconds (sd = 11.75) in the 3D point cloud view, 4.52 seconds (sd = 4.26) in the 2D first-person view, and 10.01 seconds (sd = 5.94)

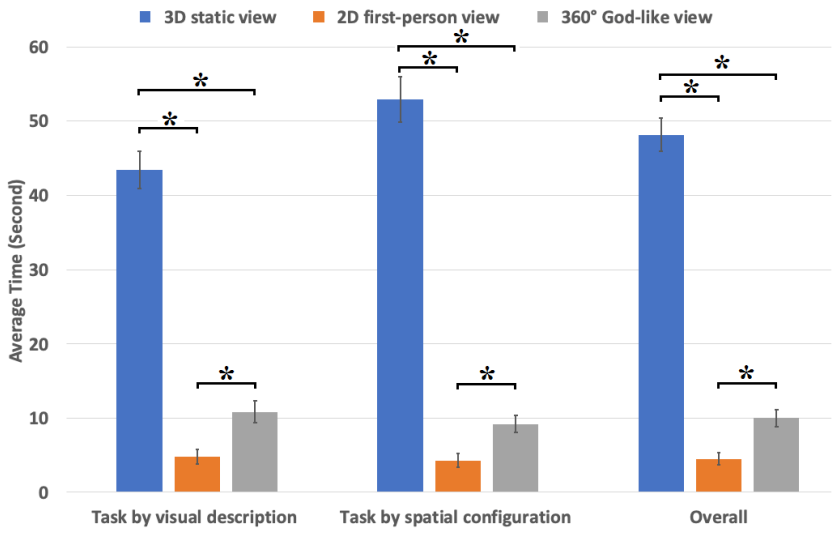


FIGURE 7.13: Average time spent on each view interface (∗: statistically significant difference)

in the 360° God-like view (Figure 7.13) for each trial task. The average duration of use showed a statistically significant difference in our device organizing task over the three view interfaces ($F(1.627, 43.932) = 325.16$, $p < 0.0005$). Post hoc analysis with a Bonferroni adjustment revealed that participants statistically spent significantly more time in the 3D static view than the 2D first-person view (43.65 (95% CI, 38.40 to 48.90) seconds, $p < 0.0005$), and more time in the 3D static view than the 360° God-like view (38.16 (97% CI, 32.81 to 43.50) seconds, $p < 0.0005$), and more time in the 360° God-like view than the 2D first-person view (5.50 (95% CI, 2.06 to 8.93) seconds, $p = 0.001$).

In our study, all of the participants tried to switch views during the study in order to take advantage of each view interface. On average, they switched 4.98 times (sd = 2.37) per trial task. No significant difference was found based on a Wilcoxon signed-rank test in the number of view switches between the two task conditions ($Z = -0.286$, $p = 0.775$) (Figure 7.14).

We recorded which view interface was used by the participants to check and confirm that the local worker correctly completed each trial. In our case, the remote expert needed to make sure the worker grabbed the right device and put it in the right location. So we recorded the view interface that was active at the end of each trial. Figure 7.15 showed the result in a total of 84 trials for each task condition (n = 28 x 3). We analyze the two task conditions individually. A chi-square goodness-of-fit test indicated statistically significant differences in the number of view interfaces that participants chose to check the final task situation for both task conditions (visual description: $\chi^2(2) = 94.57$, $p < 0.0005$; spatial configuration: $\chi^2(2) = 69.50$, $p < 0.0005$). We ran post hoc tests for pairwise

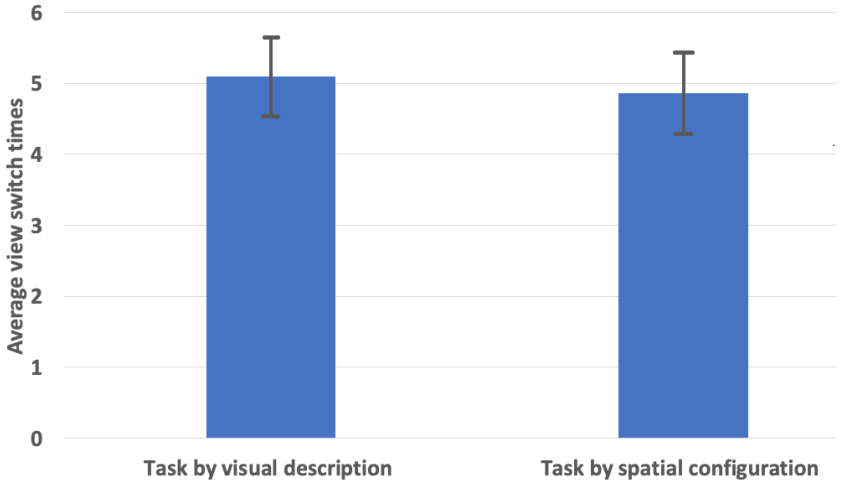


FIGURE 7.14: Average view switching times for each task condition

comparison using a Binomial test ($\alpha = 0.0167$) for each task condition, the results are shown in Table 7.4.

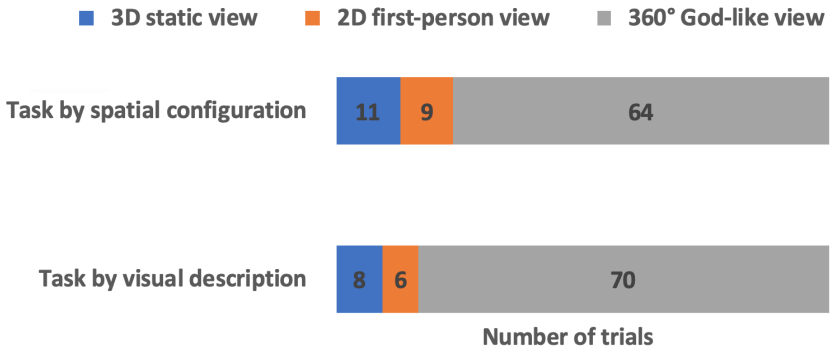


FIGURE 7.15: View interfaces participants chose to check the final task situation for each task condition ($*$: statistically significant difference)

We used a heat map to present the remote experts' movement in the shared VR space during the studies (Figure 7.16). The brighter part of the heat map indicated that the participants moved to this location more frequently. For our devices organizing tasks, participants usually started by searching for the target device on the table with all the devices, which required them to move around the table. From the heat map, we could find out that most participants chose to use the 3D static view to search for the target device. The remote experts also needed to search for the target location where the device should be placed. This behavior could cause the participants to move frequently in the shared space. According to the heat map, the 3D static view was most used for this searching process since participants made continuous and frequent movement while using this view interface.

TABLE 7.4: Pairwise comparison of the interface used to check the final task situation

	Visual description	Spatial configuration
3D static view vs 2D first-person view	p = 0.791	p = 0.824
3D static view vs 360° God-like view	p <0.0005	p <0.0005
2D first-person view vs 360° God-like view	p <0.0005	p <0.0005

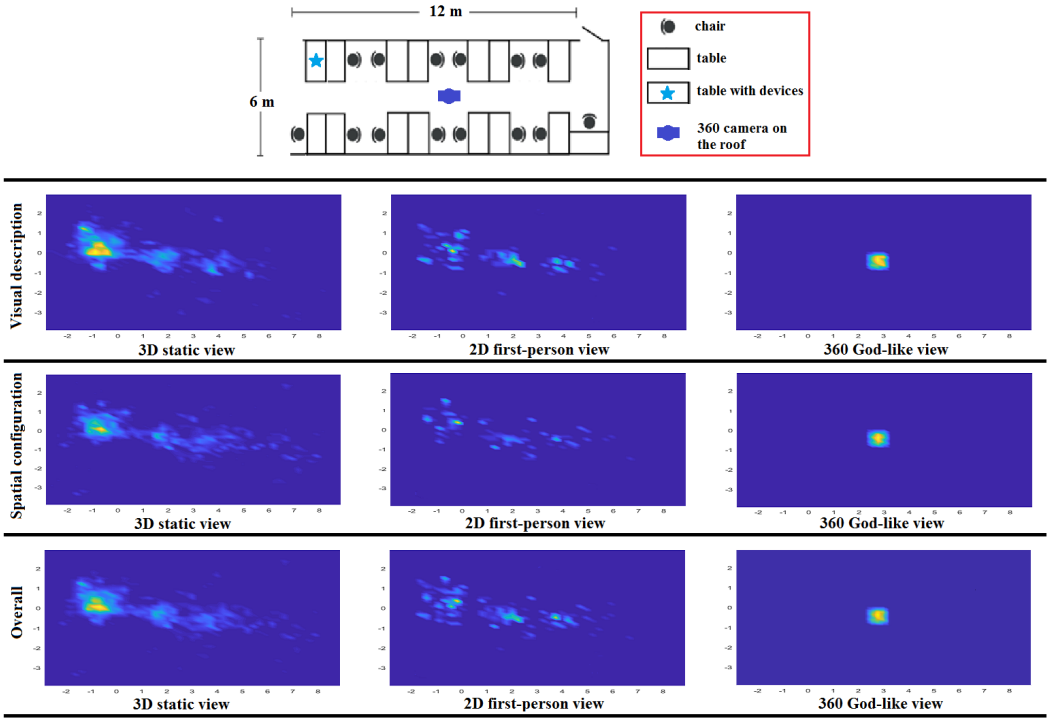


FIGURE 7.16: Heat map of remote experts' total movement in the shared VR space. The brighter part of the heat map indicated that the participants moved to this location more frequently

While one participant switched to the local worker's first-person view, his/her position also jumped to where the local worker was, then followed the local worker's movement while this view was active. We noticed that participants usually asked their partners to stand still and stare at one specific area for them to check the real-time local changes while using the 2D first-person view. After they confirmed the local situation, they usually switched the view away. Therefore, there was little movement, and the position distribution was not continuous for the 2D first-person view in the heat map.

While participants switched to the 360° God-like view, their positions were teleported to

the corresponding location where the 360° camera was in the real physical world. The viewpoint of the 360° God-like view was a fixed position. Participants could rotate their heads to observe the entire local workspace based on the panorama view without the ability to change the position of the viewpoint. Therefore, there was rarely movement in the heat map.

User experience

All of the rating questions for user experience were answered on 7-point scale items (SEQ [103] – 1: very difficult – 7: very easy; All other questions [113] – 1: strongly disagree – 7 strongly agree). A Wilcoxon signed-rank test was conducted to determine the complexity of each task condition based on participants' subjective ratings on user experience. Of the 28 participants recruited to the study, 14 participants thought searching the target location based on visual description was easier than based on spatial configuration, whereas seven participants thought the spatial configuration condition was easy and seven participants saw no difference. There was statistically significantly different in ratings of ease of searching when participants searched for the target location based on visual description (medians = 6) compared to spatial configuration (medians = 5), z = -2.174, p = 0.030 (Figure 7.17). We also found that participants enjoyed the experience significantly more while searching the target location based on spatial configuration rather than visual description (z = 2.070, p = 0.038). No significant difference was found regarding being able to focus on the task activities (z = 0.855, p = 0.392), nor on understanding their partners' focus (z = -1.070, p = 0.285).

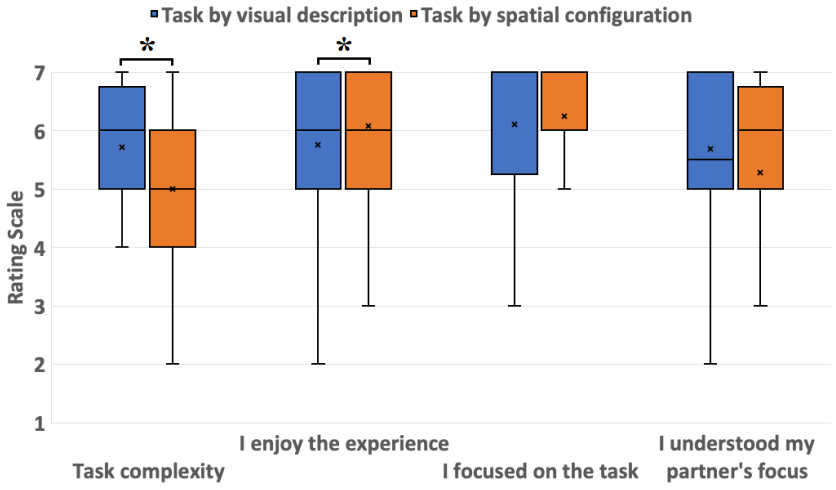


FIGURE 7.17: Results of the user experience questionnaire (∗: statistically significant difference)

Social presence

The social presence questionnaire (SoPQ) [48] we used in our study included four

sub-factors: Co-presence (CP), Attentional Allocation (AA), Perceived Message Understanding (PMU) and Perceived Behavioral Interdependence (PBI). Each of these sub-factors consisted of six closely interrelated questions scaled from 1 (strongly disagree) to 7 (strongly agree), which represented a group of Likert scale measurements. To analyze the Likert scale data, we first calculated a composite score from the six questions for each sub-factor, then calculated the mean for central tendency and standard deviation for variance [110]. As shown in Figure 7.18, compared to the neutral level rating, the results indicated that participants rated positively on CP (mean = 6.01, sd = 0.80), PMU (mean = 4.62, sd = 0.39), PBI (mean = 5.50, sd = 0.78) and overall SoP (mean = 4.94, sd = 0.49). However, participants rated negatively on AA (mean = 3.65, sd = 0.58).

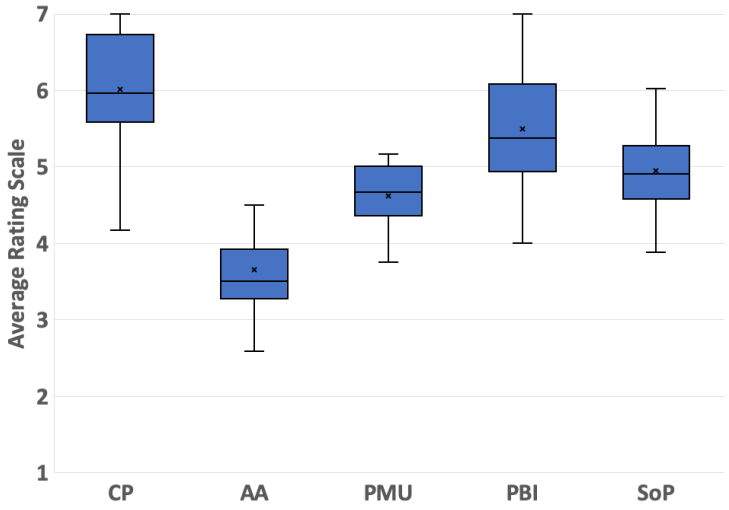


FIGURE 7.18: Results of the Social Presence questionnaire (CP: Co-presence, AA: Attentional Allocation, PMU: Perceived Message Understanding, PBI: Perceived Behavioral Interdependence, and SoP: Overall Social Presence)

Spatial presence

We used two sub-factors, Spatial Presence Self Location (SPSL) and Spatial Situation model (SSM), from SpPQ [120] to evaluate the experts' spatial presence. Each of these sub-factors consisted of four Likert rating items from 1 (Fully disagree) to 5 (Fully agree). Similar to the SoPQ, we analyzed the central tendency of the results as a whole, as well as in each sub-factors. The results showed that participants rated positively on SPSL (mean = 4.21, sd = 0.67), SSM (mean = 4.32, sd = 0.42) and overall SpP (mean = 4.27, sd = 0.48) (Figure 7.19).

System usability

In terms of SUS [13], a one-sample t-test was conducted to determine whether usability scores rated by recruited users were different from the average system usability scale

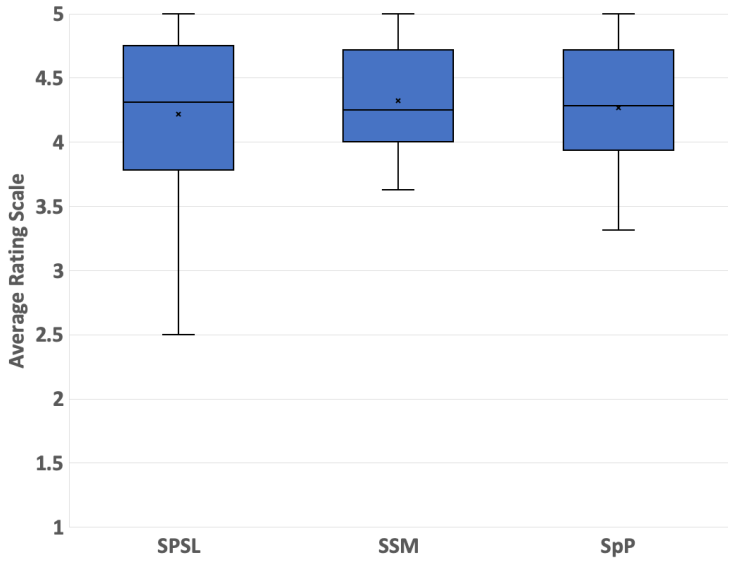


FIGURE 7.19: Results of the Spatial Presence questionnaire (SPSL: Spatial Presence Self Location, SSM: Spatial Situation model, and SpP: Overall Spatial Presence)

score of 68 [102]. Usability scores were normally distributed, as assessed by Shapiro-Wilk's test ($p = 0.670$), and there were no outliers in the data, as assessed by inspection of a boxplot. Data are mean $\pm$ standard deviation unless otherwise stated. The mean usability score (74.20 ± 13.99) was higher than the average system usability scale score of 68, a statistically significant difference of 6.20 (95% CI, 0.77 to 11.62), $t(27) = 2.343$, $p = 0.027$.

Simulator sickness

Figure 7.20 presented the average score of the Simulator Sickness Questionnaire (SSQ) [59] before and after taking our user study. SSQ consisted of 14 symptoms rated questions on a scale of (0: none – 9: severe). A Shapiro-Wilk test found that our data did not follow a normal distribution ($p < 0.0005$), so we used a Wilcoxon signed-rank test for statistical analysis. Of the 28 participants, twelve participants claimed increased motion sickness after using our system, whereas seven participants reported a decrease and nine participants felt no changes. Overall, the result showed no statistically significant increase in motion sickness symptoms after using our MR based remote collaboration system ($z = 1.797$, $p = 0.072$).

User preference

In terms of preference, participants were asked to choose one of the three views we provided as the best approach for remote collaboration according to different task objectives (Figure 7.21). A chi-square goodness-of-fit test was conducted to determine whether an equal number of participants chose each of the three view interfaces as the

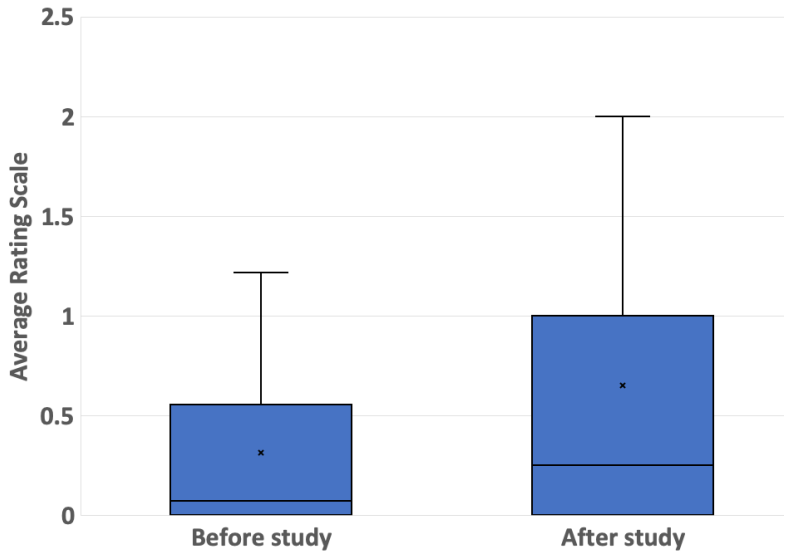


FIGURE 7.20: Results of the Simulator Sickness before and after using our MR based remote collaboration system

best based on these task objectives. The results indicated that the three view interfaces were not equally preferred by the participants for guiding ($\chi^2(2) = 19.143$, $p < 0.0005$), communication ($\chi^2(2) = 23.217$, $p < 0.0005$) and item searching ($\chi^2(2) = 22.357$, $p < 0.0005$). Over half of the participants preferred to use the 3D static view interface for guiding, communication, and item searching. Therefore, we suggest that the 3D representation of the local physical workspace played an important role in remote collaborative experiences. However, all three interfaces were equally preferred by the participants in terms of understanding the partner's focus ($\chi^2(2) = 1.786$, $p = 0.409$). We ran post hoc tests for pairwise comparison using Binomial test ($\alpha = 0.0167$) for each task objective, the results showed in Table 7.5.

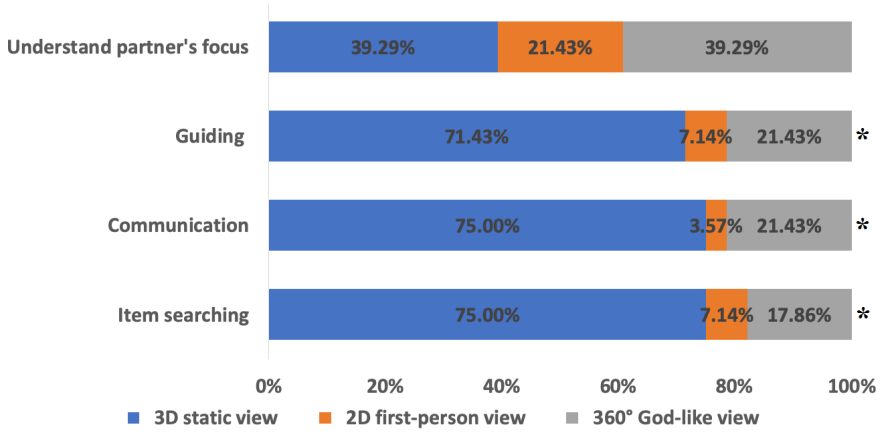


FIGURE 7.21: User preference for different task categories among the three view interfaces ($*$: statistically significant)

We also asked the participants to rate on a 5-point scale (1: strongly disagree – 5:

TABLE 7.5: Pairwise comparison of user preference for different task categories

	Item searching	Communication	Guiding	Understand partner
3D static view vs 2D first-person	p <0.0005	p <0.0005	p <0.0005	p = 0.332
3D static view vs 360° God-like	p = 0.002	p = 0.006	p = 0.009	p = 1.000
2D first-person vs 360° God-like	p = 0.453	p = 0.125	p = 0.289	p = 0.332

strongly agree) on how much they agreed with the statement "view switching is smooth" and "view switching is comfortable." The results (Figure 7.22) showed that most of the participants rated positively on view switching being smooth (median = 4) and comfortable (median = 4). One-sample Wilcoxon Signed Rank tests showed that these ratings were significantly different from a neutral level rating (smooth: Z = 2.745, p = 0.006; comfortable: Z = 2.368, p = 0.018).

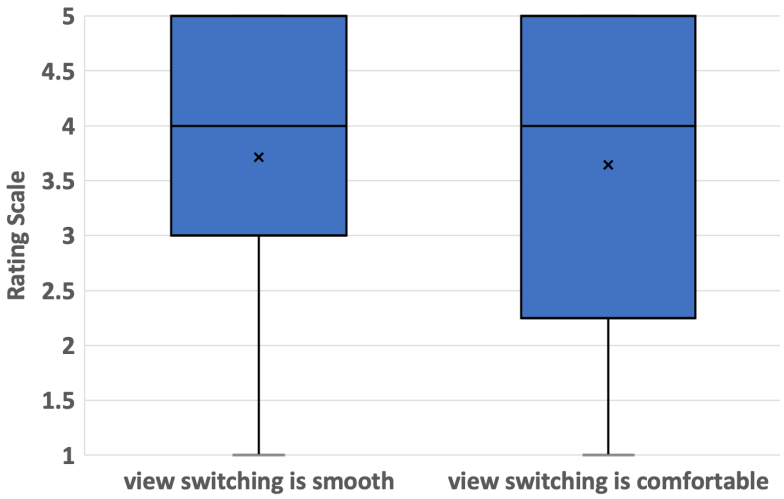


FIGURE 7.22: Rating on view switching experience between the three view interfaces

7.2.3 Discussion

Overall, the results indicated that our MR remote collaboration system's usability score was significantly higher than the average SUS score. Our system enabled the remote expert to freely navigate in a 3D VR environment to gain a better spatial awareness with the perception of the worker's actions. Therefore, participants also rated positively on overall social presence and spatial presence. However, participants ranked negatively on attentional allocation, which means that both remote and local users could easily be

distracted from their partners when other things happened. We did not find a statistically significant difference in motion sickness before and after our study. We also found that participants ranked the statements "view switching is smooth" and "view switching is comfortable" positively from the post-experiment feedback. This might be the reason why participants did not experience significant motion sickness while using our system.

3D static interface

Most of the time, participants chose to use the 3D static view interface (Figure 7.13) no matter what the task condition was. This was also supported by the user study from the first part of this chapter. We also noticed that all the participants chose to start the task process and learn the local physical layout by using the 3D static view, which verified our hypothesis **H1**: "The experts will prefer to use a global view to learn the local physical layout."

The device organizing tasks for this study mainly focused on larger-scale searching for the target item and location, which required more changes of the experts' point of view (POV). In other words, participants were required to move a lot in order to find the target. Based on the heat map (Figure 7.16), we found that participants had smooth movement while using the 3D static view. On the other hand, they rarely moved while using the other two view interfaces. Therefore, we believed that participants mainly used the 3D static view for target searching in our study. Furthermore, a majority of participants (75%) also chose the 3D static view as the best approach for item searching based on user preference ranking (Figure 7.21). In this case, we could verify our hypothesis **H2**: "The experts will prefer to use a global view to search for the targets."

We observed that all of the participants chose to make some annotations in the VR space by using the 3D line drawing function to indicate the target device and target location, together with words like *"Grab the item which I marked"* or *"Put the headset on the table where the arrow points to."* We also noticed two participants even tried to draw lines from where the worker stood to the target location to show the moving path for the worker. The line drawing function only worked for the 3D static view since it was a challenge for us to correctly project 2D annotations into the 3D space while using the other two view interfaces. Therefore, in our study, participants mainly chose to use the 3D static view to guide the local worker. User preference ranking on guiding also supported this finding, which verified our hypothesis **H3**: "The experts will prefer to use a global view to guide the local worker."

In terms of user preference, besides item searching and guiding, participants also showed

their preference for communication while using the 3D static view in this study (Figure 2.21). This is different from what we observed in part one, in which 55% of the participants preferred to use 2D first-person view for communication. The device organizing tasks required more changes of the experts' POV for large-scale item searching, whereas the label reading task from the last study focused on small-scale manipulation with a high-resolution focused view. Prior work [71] indicated that the POV controlled by the expert improved user performance while dealing with tasks that required more changes of the POV. The 3D static view supported fully independent viewpoint control for the experts, and participants tended to show their preference for the approach which could provide more help for completing the tasks. Therefore, they ranked the 3D static view interface as the best, not only for item searching and guiding but also for communication in this study. In this case, the best view interface for communication was task dependent.

From the user study in part one of this chapter, we learned that the resolution and quality of the point cloud captured from a depth sensor could not handle a large-scale workspace. In this study, we updated our 3D static scene from a single point cloud set to a combination of point cloud and 3D mesh models. The result showed that participants spent much more time with the 3D static view than the other two (Figure 7.13). However, there were still some limitations for our current 3D static view. For example, participants mentioned, *"The table models are not exactly the same as the real table in the office"*, and *"It is unable to move objects in the 3D global view"*.

2D first-person view

The heat map (Figure 7.16) indicated that participants tried to use the 2D first-person view throughout the whole task process since their positions were spread over the map. The first user study in this chapter showed that a real-time first-person view was useful for checking detailed information of one small focused area. However, the device organizing task did not require the same high-resolution focus view as the previous label reading task. Participants also pointed out that *"In theory, it should work well, but sharing a person's view easily leads to nausea"* and *"the Magic Leap camera's position is not the same as my partner's eyes, so I don't always see what he was looking at from the first-person view."* The first-person view was unstable and not natural enough, so participants quickly switched away from this interface. Therefore, the position distribution was not continuous for the 2D first-person view on the heat map, and participants spent less time on this view interface (Figure 7.13).

Based on the finding of the first user study, we found that participants chose to use

the 2D first-person view to check the local task situation. In this study, we intended to check if the participants preferred to use a real-time high-resolution view to check the local situation (**H4**). However, from Figure 7.15, we knew that participants in fact chose mostly to use the 360° God-like view to check if the worker completed the task correctly. Although we supported 4K resolution for the 360° God-like view, the resolution of this view was still smaller than the 2D first-person view in the same FOV. Therefore, we failed to verify our hypothesis **H4**: "The experts will prefer to use a real-time high-resolution view to check the local situation in a focused area ." Participants may have used the 360° God-like view instead of the 2D first-person view to check local task situation because it provided view independence. Participants even said that *"My partner did not cooperate when I asked him to watch the targets in the first-person view, so I can only use the 360° view to check the local situation"* and *"I need to talk more to ask my partner to watch the target area."*

360° God-like view

From the heat map, we found that participants rarely moved after switching to the 360° God-like view since they could only rotate their heads to observe the captured panorama view of the local workspace. From Figure 7.15, we also found out that participants preferred to use this view to check the local situation. We aligned the orientation of the 360° God-like view with the 3D static view. While participants switched the view from 3D to 360°, they experienced the viewpoint jumping. However, they still watched the same direction with independent view control, which enabled them to quickly relocate to the area they would like to check. The view independence could have played a significant role for participants choosing to use the 360° God-like view to check the local situation, as shown by prior work [73] [113].

Visual Condition vs Spatial Condition

The results indicated that participants spent significantly less time on completing the task when the target locations were described by visual cues in color. The subjective user experience feedback verified this. Participants thought that searching the target location by visual cues was significantly easier than spatial configuration. For Visual Condition, they reported that *"The color markers are more obvious for me."* For Spatial Condition, they emphasized that *"I usually count the tables at least twice in case I do not make mistakes."* We tested the time spent on each view between these two task conditions, and only found a significant difference for the 3D static view. The difference between these two conditions was how to search for the target location, and participants mainly used the 3D static view for searching. In this case, there was an increase in the time spent on searching for the Spatial Condition. We believe that searching for the target using the

spatial configuration might increase the participants' cognitive load, which caused an increase in task completion time. However, participants also showed their preference in searching for the target by spatial configuration rather than visual description. We noticed that participants usually stood still and counted the tables while searching by spatial configuration, whereas they needed to move a lot while searching by visual description.

Local user feedback

Overall, local users thought that observing the remote guidance overlaid on top of the real world was novel and helped them to complete the task. They reported that *"The annotation was quite straightforward and accurate for me to understand the guidance"* and *"Seeing the virtual head of the remote user made me feel like he was actually moving around me"*. However, they also pointed out that the 3D annotations sometimes might not be correctly aligned with the target object. The local and remote users observed the targets from different locations and directions, which could lead to the visual misalignment in the 3D space if the annotation was far from the targets. We asked the experts to draw the annotations as close as possible to the target to reduce this visual dislocation.

Local users also reported that they usually followed their partners' movements to check the possible annotation cues. However, if the remote experts teleported in the VR space, the head avatar used to present them in the local space would jump to another location. Therefore, the local users might lose where their partners were and had to relocate them. They suggested that adding a visual effect to show the path of teleportation would help them increase task performance.

7.3 Conclusion

In this chapter, we present two user studies to analyze the remote experts' behaviors while using an MR based remote collaboration system. The result showed that remote experts prefer to use a global view to learn the local physical layout, search for the targets, and provide guidance. We also found that the experts chose to use the 360° live view with independent view control rather than the 2D high-resolution first-person view to control the task procedures and check the local worker's actions.

Based on the user behavior analysis from this chapter, we could summarize some interface design guidelines for room-scaled remote collaboration:

- A global view, such as capturing and sharing the entire local workspace as an

integrated 3D scene, can increase the expert' spatial awareness and support view independence, which is better for the remote expert to learn the local physical layout and search for the targets.

- A live view with view independence, such as the 360° live God-like view, can help the remote experts to check the local situation and control the task process.
- A high-resolution live view, such as the 2D live first-person view, shows an advantage in supporting detailed manipulation.
- A virtual representation of the remote partner in the shared VR environment can effectively indicate to users where their partner is, which is helpful for them to track the partner's action and control the task process;
- World-stabilized 3D line drawing supports the experts in a variety of ways to show guidance cues, such as indicating the targets, leading the moving path, and even writing text instruction in space.

From the second study in this chapter, we still found some limitations of our MR based remote collaboration system. For example, the shared 3D scene did not support object segmentation; therefore, it was unable for the experts to move the virtual target in the VR world to support more intuitive guidance. Furthermore, our system setup only enabled remote collaboration in an indoor environment because the 3D capture devices required cables linked to one powerful PC. In the next chapter, we discuss how we re-implemented our MR remote collaboration system on mobile devices, enabling people to use it anywhere and anytime. In addition, we tried to build a 3D triangulated mesh from the captured 3D point cloud to increase the resolution of the shared scene.

Chapter 8

Mobile Based Remote Collaboration

The scene capturing and sharing remote collaboration system I introduced in previous chapters required VR headsets connected to the PCs, which could only be tested in indoor environments. In order to support remote collaboration in a common and real working scenario, in this chapter, we implemented our MR remote collaboration system on mobile devices that enabled an expert to provide remote real-time assistance anytime and anywhere.

By using the Google ARCore [1] position tracking, we could integrate the keyframes captured by one external depth sensor attached to the mobile phone (Samsung Galaxy s8) into one single 3D point-cloud data set to create a copy of the local physical environment. This captured local scene was then wirelessly streamed to the remote side for the expert to view while wearing a mobile VR headset (HTC VIVE Focus [2]). In this case, the remote expert could immerse himself/herself in the VR scene and provide guidance while feeling like sharing the same work environment as the local worker.

8.1 System Setup

In this chapter, we ported our previous desktop-based system from chapter 4 to mobile devices to improve the usability and user experience (Figure 8.1). Instead of wearing a cumbersome headset connected to a PC, the expert only needed to use a light-weight self-contained headset (the HTC Vive Focus) during the tasks. At the same time, the display device on the local side was switched from a VR headset to an Android smartphone [3].

The prototype of our proposed mobile collaboration system was subdivided into three

[1] https://developers.google.com/ar/
[2] https://enterprise.vive.com/us/vivefocus/
[3] Video link for mobile setup: https://www.youtube.com/watch?v=GRLVEShl-wY

logical sub-systems: (1) the local capturing and viewing system which supported the worker on completing the task (Figure 8.1A), (2) the remote guiding and viewing system which enabled the expert to provide guidance (Figure 8.1C), and (3) a PC server was responsible for point cloud fusion and mesh triangulation.

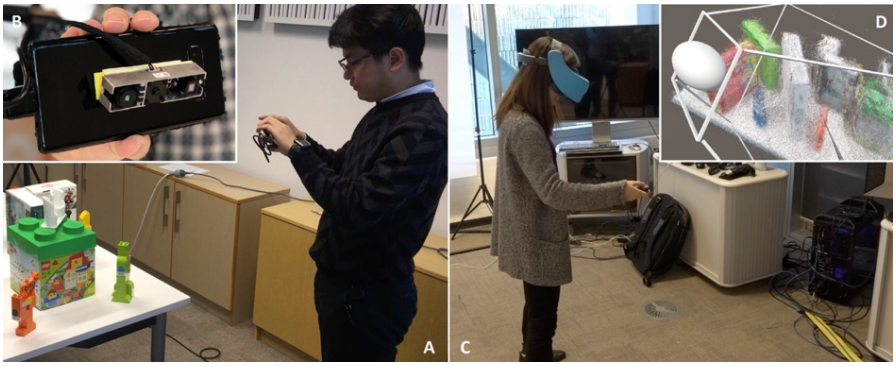


FIGURE 8.1: The mobile-based MR remote collaboration system. A: The local worker held a mobile phone to capture the local scene and observe the remote guidance via the display. B: The mobile phone with a depth sensor attached on the back for scene capture. C: The Remote expert observed the captured local environment in VR and provided guidance with a controller. D: The Remote expert's view with static point cloud background and local view frustum enabled in the VR display.

To capture the local scene, we attached a depth sensor (the Orbbec Astra Mini long-range sensor [4]) to an Android smartphone (Figure 8.1 B). While the local worker held the phone and walked around, the system could capture a set of 3D point cloud data from different locations. Using the ARCore position tracking solution, these single-frame point clouds could be transferred into the same VR world coordinate system. Once captured, these single-frame point clouds were wirelessly streamed to a desktop PC, which acted as one server. On the server side, the 3D point clouds would be stitched together into one integrated single point cloud set (Figure 8.2) using the same keyframe-based registration method from our PC-based remote collaboration system introduced in Section 4.4.

If the current frame was detected as one keyframe, the server would automatically fuse this keyframe with the previous keyframes and wirelessly stream the data of this new keyframe to the remote side. In this case, the remote system was only responsible for rendering one keyframe at a time. We restricted the color and depth maps received from the depth sensor to a resolution of 320 * 240 pixels. Therefore, at one time, the system only processed no more than 320 * 240 points (noise points were deleted based on pre-defined depth boundary). Our system could achieve 30 fps on both the local and the remote side, and 20 fps on the sever.

[4] https://orbbec3d.com/product-astra-pro/

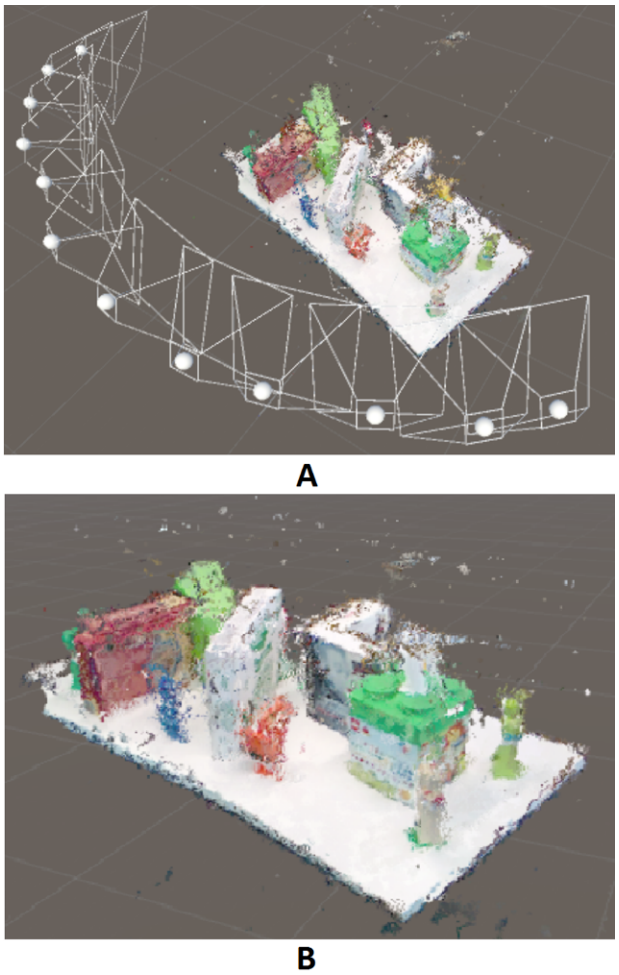


FIGURE 8.2: A captured scene. A: Keyframe registration based on a series of related frames (10-20 frames); B: After registration the entire scene was captured as 3D point cloud

The remote expert could view the captured local scene from the self-contained VR display on the remote side. The Vive Focus world-scale tracking enabled the remote expert to freely move around in the 3D VR world while navigating to the target location. Furthermore, we tracked the local mobile phone position with the support of the ARCore library and rendered it as a view frustum in the remote VR world. In this case, the remote expert could observe the partner's movement and focus during the guiding process, which reproduced the same natural environment as two people working face-to-face in the real world. One limitation of this setup was that the mobile phone was held in the worker's hand, so the mobile phone position was not the same as the position of the local worker's head.

The scene capturing needed to be processed before the task started, so it could not show real-time changes during collaborative tasks. To solve this issue, we also streamed real-time 2D video captured by the local mobile phone to the remote side via the PC

server, and displayed it together with the worker's view frustum in the remote expert's VR view (same as the TPV interface setup in Chapter 6). During the collaboration task, the remote expert could learn the local workspace's spatial layout and search for target objects by using the background 3D point cloud scene and provide detailed guiding cues by using the live 2D local video.

Besides, the PC server could render the captured local point cloud into one integrated triangulated mesh (Figure 8.3) to support physical interaction for the remote expert [5]. The triangulation process started after the scene capture process completed on the server. The point clouds from all the keyframes were integrated as one single data set. Using the Point Cloud Library [6] (PCL library), we could calculate the triangulated mesh for this data set. The processing time depended on the number of keyframes we saved. It usually took around five seconds to transfer the point cloud data contained 15 to 20 keyframes into one triangulated mesh. The PC server streamed the triangulated mesh to the remote side once it was created. By using this mesh, we could support more interaction cues for the remote expert. For example, the remote expert could put a virtual ball on top of the table mesh, and the local worker would see this ball as an AR element overlaid on top of the real table through the mobile phone display. In the next step, we intended to attach high-resolution textures to this triangulated mesh. In this case, instead of using point cloud, we can present the local scene as one textured mesh with much more details for the remote expert.

During the task process, the local mobile phone was used as a video see-through display. To provide direct guidance, pointing information from the remote side was also streamed back to the local side as AR cues overlaid on top of the local video. The current Vive Focus controller only had 3DOF orientation tracking; therefore, it could not be used as a direct pointer in the VR space. When the expert pressed the trigger on the controller in our system, a virtual laser ray was rendered from the top of the controller model. By rotating the controller in hand, the expert could use this laser ray to point to the target object.

For the PC server, we used a PC set up with an Intel Core i7, 32GB RAM, and NVIDIA GeForce GTX 1060 GPU, running Windows 10. This local PC was responsible for point cloud fusion, mesh triangulation, and switching data between the local and remote systems. We used a network socket to wirelessly stream data based on UDP connection

[5] Video link of mesh triangulation: https://www.youtube.com/watch?v=GZRnyBH16rc
[6] http://pointclouds.org/

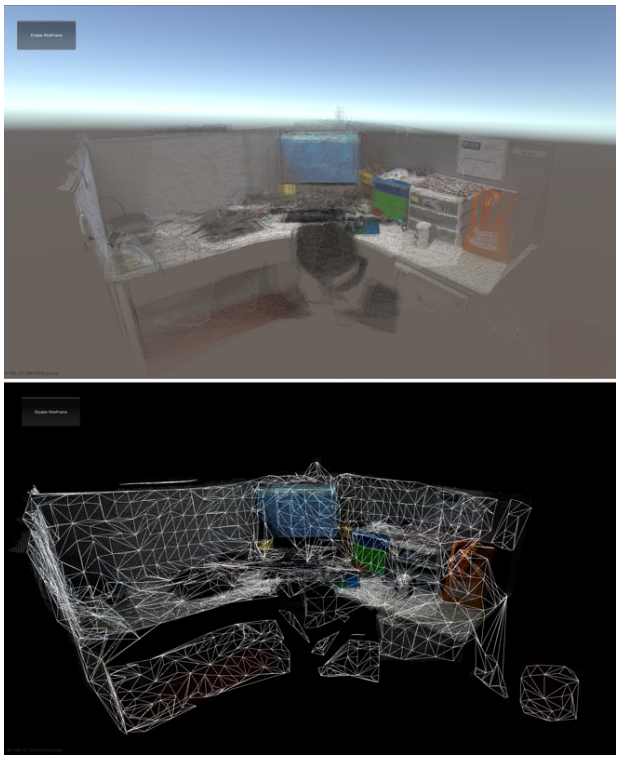


FIGURE 8.3: The triangulated mesh built from the point cloud of the local scene

via the NETGEAR Nighthawk X6 WiFi Router. The data exchange could achieve real-time. Unity engine was used for the scene rendering, and both the local and remote systems could run at 30 fps during the collaborative process.

8.2 The Collaborative Task Process

Figure 8.4 shows the data flow during different steps of our mobile-based remote collaborative experience. To complete a collaborative task, the local worker held the mobile phone with the depth sensor attached and first scanned the local physical workspace before the collaboration task started. The point cloud captured could be seen on the smartphone display and shared with the remote expert in real-time using wireless data streaming via the PC server. Once the entire local workspace was captured (as judged by the local worker), he/she could press a button on the touch screen to finish the scene capture process. The server would then automatically combine the point cloud set into one triangulated mesh and stream this mesh information to the remote side.

After the local scene was captured and shared as a 3D point cloud between the local and remote users via the PC server, the remote expert could start guiding the local worker to complete the target task goal (Figure 8.1). At this stage, the local mobile phone screen

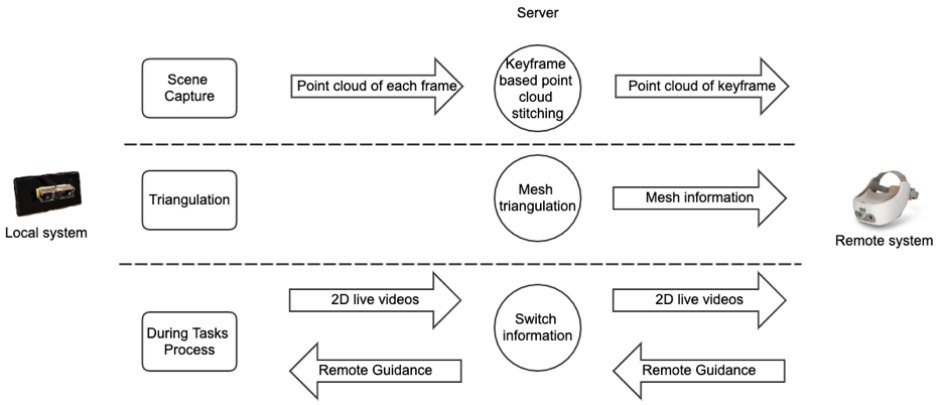


FIGURE 8.4: Data flow from local system to remote system via the server

would switch into a video see-through display to show the remote AR guidance cues overlaid on top of the real-time 2D local video. By holding the smartphone with one hand to check the remote guidance cues, the local worker could use the other hand to interact with the physical objects in the real work environment to achieve the task goal.

On the remote side, the expert could use the hybrid interface displayed in the self-contained VR headset that combined the 3D static point cloud background with the 2D real-time foreground video to control the task process. This interface was designed the same as the third-person view (TPV) interface introduced in Chapter 6, which showed the advantage in enhancing the perception of co-presence. A static 3D virtual point cloud of the local scene was displayed as a background in the remote expert's VR world. Real-time 2D video of the local worker's view was attached to the local worker's view frustum as a 2D window. By holding the controller and pressing the trigger, the remote expert could point to the 3D virtual objects in the VR world with the laser ray. Simultaneously, verbal communication was also enabled for the remote expert in controlling the guidance process. Furthermore, since the triangulated mesh was applied and overlapped with the 3D point cloud background, the remote expert could also place pre-defined 3D virtual objects on top of the mesh to provide alternative cues besides pointing guidance.

8.3 Conclusion

In this chapter, we developed a prototype MR remote collaboration system using mobile devices, which could provide users with a more natural and convenient way to collaborate with each other. Compared to our previous MR remote collaboration system introduced in Chapter 4, this mobile system could be used without space limitations, especially for the local worker. In this case, the local worker could ask a remote expert

to provide MR based guidance in an outdoor working environment by holding just a mobile phone. In other words, we tried to bring our remote collaboration system to a real working scenario by using the setup with mobile devices.

Chapter 9

Conclusions and Future Work

This Ph.D. research explored how to use Mixed Reality technology to enhance the performance of remote collaboration in a room-scale workspace. In particular, I focused on discussing the advantages and limitations of enabling remote collaborators to share a 3D view of the same workspace simultaneously from different physical locations. A 3D reconstruction of the physical work environment is an ideal approach for showing the basic physical layout in a shared VR world. It can provide the experts with a way to have an independent viewpoint control from each other. Therefore, I chose to use 3D reconstruction in an MR remote collaboration system to deal with the issues of room-scale remote collaboration.

To conduct my research, I developed an MR remote collaboration system that combined a low-resolution static 3D reconstruction of the environment surrounding the local worker with different approaches to show real-time local views. I evaluated my design in five user studies with different interfaces and communication cues. First of all, I focused on enhancing view independence and spatial awareness for the remote expert with an oriented 3D view interface in a model assembling task (Chapter 3). Next, I extended my MR remote collaboration system to enable it to capture the entire local scene as a 3D dense point cloud set in a room-scale workspace (Chapter 4). I used simple guiding tasks to compare the performance of this system with the previous single-frame 3D capture system (Chapter 5). I further combined the static 3D reconstruction with real-time feedback to enhance the collaborative experience and designed several interfaces to present this combination (Chapter 6). Finally, I evaluated the user behavior while using my MR remote collaboration system design, and tried to explore how my interface design could better support the collaborative tasks (Chapter 7). I also extended my MR remote collaboration system from desktop PC to mobile devices, enabling remote guidance to be provided anywhere and anytime (Chapter 8). In this case, my system

could support remote collaborative tasks in a real working scenario.

In this final chapter, I summarize my research results into a list of contributions followed by directions for future work. These findings can help provide some high-level guidelines for researchers who intend to work on room-scale remote collaboration systems using AR and VR technologies.

9.1 Contributions

This Ph.D. research provides some of the first studies conducted to evaluate the task performance and user experience of MR based room-scale remote collaboration systems on assembling and searching tasks. Many previous studies have been conducted within a small workspace, but there is a need for such studies that consider design issues while working in a room-scale workspace. My study provides some possible insights into this research area.

The primary hypothesis of my research is that MR based remote collaboration systems can support better task performance and user experience for room-scale collaborative tasks (**Q1** and **Q2**). The results of my study confirm this hypothesis. My research had the following key findings:

- The combination of single-frame point cloud capture and orientation information can significantly increase the perception of co-presence (Chapter 3) because the remote experts always need to rotate their heads to catch the local view, which strengthened the perception of their partner's presence.

- Capturing and sharing the entire room-scale workspace as one static VR scene enables the remote expert to virtually walk through the local workspace and observe the local objects in 3D from a range of perspectives, which significantly reduces task completion time for target searching. Remote experts preferred searching the target using the 3D replica of the local workspace. However, remote experts did not feel confident in completing the task goals without real-time local feedback (Chapter 5).

- Showing the real-time local view as a picture-in-picture window (FPV) to support real-time feedback could significantly reduce the physical stress during the task process and is most preferred by remote experts. In contrast, attaching the real-time local view to the worker's view frustum (TPV) significantly increases both physical and mental stress but supports a better perception of co-presence. Furthermore,

displaying the real-time local view as a dynamic point cloud (PCV) tends to be most difficult to use compared to FPV and TPV (Chapter 6).

- Remote experts showed positive feedback for social presence and spatial presence while using the switching view MR interface that combined a 3D static view, a 2D first-person view, and a 360° God-like view. The system usability score was significantly higher than the average usability score of 68. Furthermore, remote experts did not feel significant motion sickness while using this interface. The view switching process was confirmed to be comfortable and smooth compared to the neutral level rating (Chapter 7).

I also confirmed that the combination of different remote collaboration media (3D, 2D, 360°) complement each other for room-scale collaboration (**Q3**). When I combined different remote collaboration mediums together, I found that:

- Remote experts chose to use the 3D static view to learn the local physical layout and search for the targets (Chapter 7, part 2). The 3D static view supports a global view for the experts to experience the local physical world's spatial layout with the freedom to navigate themselves to any location of the shared scene.

- Remote experts chose to use the 360° live view with independent view control rather than the high-resolution 2D first-person view to control the task process and check the worker's actions (Chapter 7, part 2).

- If the collaborative task required detailed manipulation on the local objects, remote experts showed their preference for using a 2D first-person high-resolution view (Chapter 7, part 1).

Finally, based on user behavior analysis, I conclude with some high-level guidelines for MR remote collaboration systems designers in a room-scale workspace (**Q4**):

- A global view, such as capturing and sharing the entire local workspace as an integrated 3D scene, can increase the expert' spatial awareness and support view independence, which is better for the remote expert to learn the local physical layout and search for the targets.

- A live view with view independence, such as the 360° live God-like view, can help the remote experts to check the local situation and control the task process.

- A high-resolution live view, such as the 2D live first-person view, shows an advantage in supporting detailed manipulation.
- A combination of different remote collaboration mediums can complement each other, but the ways we combine the mediums may affect the room-scale remote collaboration in different aspects. For example, the FPV supports small local view but reduces physical stress, while the TPV enhances co-presence but increases physical and mental stress.
- A virtual representation of the remote partner in the shared VR environment can effectively indicate to users where their partner is, which is helpful for them to track the partner's action and control the task process;
- World-stabilized 3D line drawing supports the experts in a variety of ways to show guidance cues, such as indicating the targets, leading the moving path, and even writing text instruction in space.

9.2 Future Research Directions

My MR based remote collaboration system, which combined 3D capture hardware, co-presence techniques, and efficient guidance cues, conveyed a similar communication experience remotely as in face-to-face collaboration in a room-scale workspace. However, there are still many limitations that I can point out in the system design.

The static 3D capture of the local environment cannot support complicated task scenarios, such as the tasks that require frequent local changes or several workers to collaborate together. Frequent local changes would cause the captured static 3D scene and the real local physical environment to become different from each other with the task process going on. Furthermore, if there are several workers working in the same workspace, it would further decrease the remote expert's spatial awareness of the local space and increase the difficulty of managing the task process. In this case, during my research studies, I focused on analyzing the object searching and organizing tasks that make minimal changes to the local environment and limited one worker working in the local workspace.

Besides, the captured 3D point cloud can only show the general appearance of the local objects due to the low depth resolution supported by current depth sensors. It is hard for the remote expert to identify small objects and partly captured objects in the shared VR scene. Therefore, the target objects defined in my studies were always big enough

for the remote expert to search for, which limited my research to cover more common task scenarios.

To deal with these issues, I can point out some key future research directions in remote collaboration system design in a room-scale workspace.

Multi-sensors based Dynamic 3D Capture

The depth sensor can acquire 3D information of the local environment, and then reconstruct and share the local scene with the remote expert to improve their spatial awareness. This allows them to understand the spatial layout of objects better, have improved depth perception, handle occlusion in the scene, and navigate the scene as freely as in face-to-face collaboration. However, this method has its own limitations. For example, a static mesh cannot support real-time updates of any local environmental changes, and most live point cloud scenes have a small field of view and interactive volume.

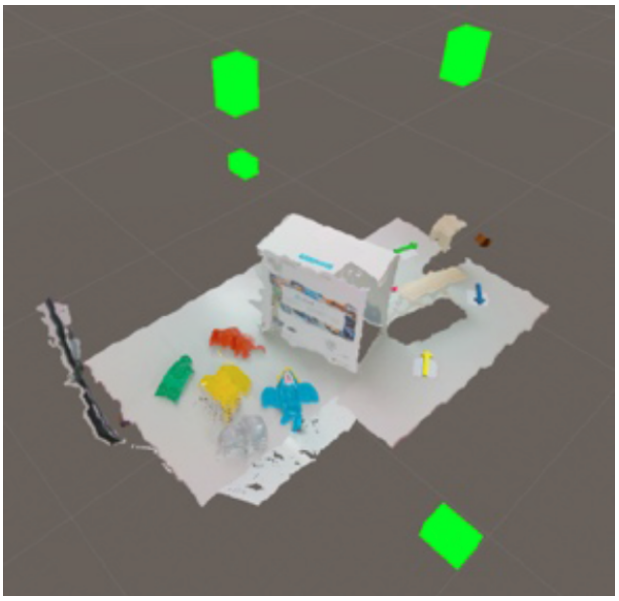


FIGURE 9.1: A stitched point-cloud scene, with the locations of four depth cameras indicated by green boxes

To address the above limitation, one approach is to enable live streaming of a dense 3D point cloud model captured from the local worker's environment. The live dense scene can be constructed by stitching together multiple depth sensors' point-cloud frames synchronously in real-time, and any changes in the environment can be updated to the remote expert in real-time in 3D. There have been very few studies [69][60] that used multiple synchronized depth cameras to stitch and capture a large field of view with live point cloud for remote collaboration. Figure 9.1 shows early results from a system that could achieve this. In this system, the local workspace was captured by four

depth sensors with their point-cloud outputs stitched together in real-time, and then wirelessly streamed to the remote side. This 3D dataset was then rendered in a dense scene in a VR environment for the remote expert, which could respond in real-time to any scene changes. To calibrate and stitch all four sensors correctly in the space, we could perform a one-time calibration using the Multiple-camera System Calibration Toolbox for MATLAB [77]. In a future study, I would like to investigate this 3D capturing and rendering approach to test how it could be used to enhance remote collaboration.

Increasing Mutual Awareness with Real-time Skeleton Tracking

One of the significant features of face-to-face collaboration is that users can directly see each other, which significantly increases the communication efficiency and social presence. To achieve this and go one step beyond environment capturing, I intend to track the user's whole body movement and use it to control a 3D avatar in the VR world. In this case, users can see their partner's whole body instead of the head avatar that I introduced in my MR remote collaboration system setup.

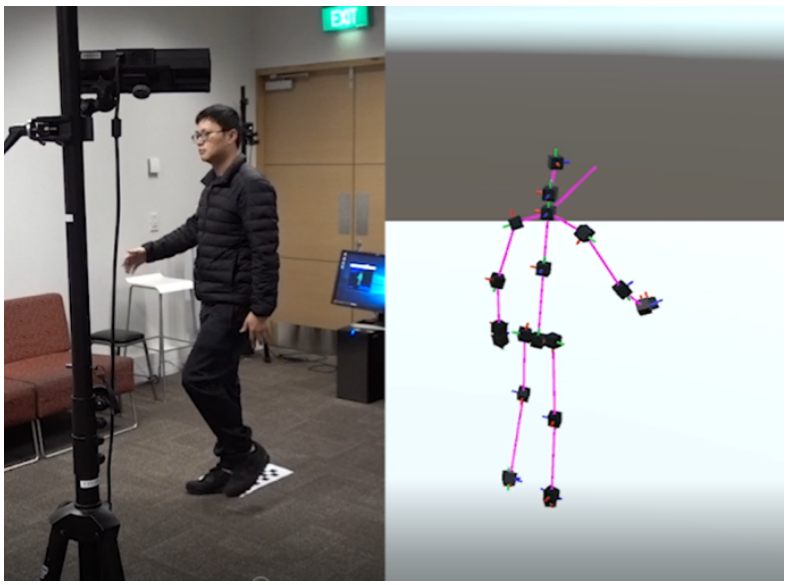


FIGURE 9.2: Skeleton tracking: one user stands in the middle of three Kinects (left); fused skeleton data (right)

Real-time full-body tracking in VR is important for providing realistic experiences, especially for applications such as remote training, education, and social VR. The Microsoft Kinect v2 depth sensor can provide skeleton data for a user in real-time. However, due to occlusion issues and front/back ambiguity errors, one Kinect is not always reliable enough for the correct capture of 360° full-body movements. One client-server setup with multiple Kinect v2 cameras can ideally solve this issue. As shown in Figure 9.2, the user's skeleton data could be tracked by three individual Kinects and fused together [122]. A ray from the neck joint was used to indicate the user's facing direction in

real-time. Based on the facing direction, we knew that the user raised his right arm.

In the future, I intend to integrate this full-body skeleton tracking system into my MR based remote collaboration system to present a remote partner as a virtual avatar, and compare it to a face-to-face collaborative process to see how well we can convey the same experience.

Semantic Segmentation and 3D Scene Parsing

Semantic segmentation can be used to recognize objects with assigned labels in the 2D images using convolutional neural networks (CNNs) [80]. As shown in Figure 9.3, Yabe et al.'s work showed how to segment objects from a dense point cloud set by combining semantic segmentation and visual SLAM tracking [123]. In this case, the remote expert can interact with the shared 3D scene by moving the virtual target objects in the VR environment. In the future, I intend to investigate how this 3D scene parsing technique can be used in a remote collaboration system to support a more natural and efficient collaborative experience.

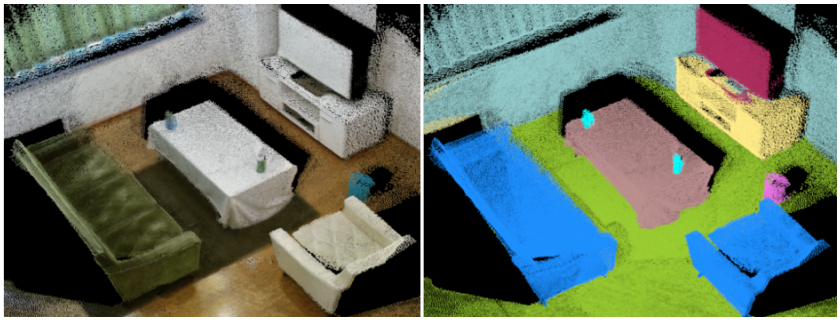


FIGURE 9.3: 3D parsing system: Original point cloud set (left); Parsing result (right) [123]

In the future, I would like to combine the dynamic 3D scene capture, real-time skeleton tacking, and object segmentation techniques to enhance the task performance and user experience for remote collaboration in a room-scale workspace. Overall, I intend to reproduce the face-to-face collaborative experience more naturally and intuitively using advanced Mixed-Reality techniques.

Bibliography

[1] Matt Adcock, Stuart Anderson, and Bruce Thomas. "RemoteFusion: real time depth camera fusion for remote collaboration on physical tasks". In: *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*. ACM. 2013, pp. 235–242.

[2] Jörgen Ahlberg. "A system for face localization and facial feature extraction". In: (1999).

[3] Leila Alem and Jane Li. "A Study of Gestures in a Video-mediated Collaborative Assembly Task". In: *Adv. in Hum.-Comp. Int.* 2011 (Jan. 2011), 1:1–1:7.

[4] Judith Amores, Xavier Benavides, and Pattie Maes. "Showme: A remote collaboration system that supports immersive gestural communication". In: *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM. 2015, pp. 1343–1348.

[5] Huidong Bai, Lei Gao, and Mark Billinghurst. "6DoF input for hololens using vive controller". In: *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*. ACM. 2017, p. 4.

[6] Martin Bauer, Gerd Kortuem, and Zary Segall. "" Where are you pointing at?" A study of remote collaboration in a wearable videoconference system". In: *Digest of Papers. Third International Symposium on Wearable Computers*. IEEE. 1999, pp. 151–158.

[7] Paul Beddoe-Stephens. *New Publisher Tools for 360 Video*. 2016. URL: https://www.facebook.com/facebookmedia/blog/new-publisher-tools-for-360-video.

[8] Mark Billinghurst and Hirokazu Kato. "Collaborative mixed reality". In: *Proceedings of the First International Symposium on Mixed Reality* (1999), pp. 261–284.

[9] Mark Billinghurst, Alaeddin Nassani, and Carolin Reichherzer. "Social panoramas: using wearable computers to share experiences". In: *SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications*. ACM. 2014, p. 25.

[10] Sébastien Bottecchia, Jean-Marc Cieutat, and Jean-Pierre Jessel. "TAC: augmented reality system for collaborative tele-assistance in the field of maintenance through

internet". In: *Proceedings of the 1st Augmented Human International Conference* (2010), p. 14.

[11] Jason Brand and John S Mason. "A comparative assessment of three approaches to pixel-level human skin-detection". In: *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000* 1 (2000), pp. 1056–1059.

[12] Scott Brave, Hiroshi Ishii, and Andrew Dahley. "Tangible interfaces for remote collaboration and communication". In: *Proceedings of the 1998 ACM conference on Computer supported cooperative work - CSCW '98* (1998), pp. 169–178.

[13] John Brooke. "SUS: a ¡°quick and dirty'usability". In: *Usability evaluation in industry* (1996), p. 189.

[14] Erik Bylow, Jürgen Sturm, Christian Kerl, Fredrik Kahl, and Daniel Cremers. "Real-time camera tracking and 3D reconstruction using signed distance functions." In: *Robotics: Science and Systems* 2 (2013).

[15] Jean Carletta, Robin L Hill, Craig Nicol, Tim Taylor, Jan Peter De Ruiter, and Ellen Gurman Bard. "Eyetracking for two-person tasks with manipulation of a virtual world". In: *Behavior Research Methods* 42.1 (2010), pp. 254–265.

[16] Rohan Chabra, Adrian Ilie, Nicholas Rewkowski, Young-Woon Cha, and Henry Fuchs. "Optimizing placement of commodity depth cameras for known 3D dynamic scene capture". In: *2017 IEEE Virtual Reality (VR)*. IEEE. 2017, pp. 157–166.

[17] Jeff Chastine, Kristine Nagel, Ying Zhu, and Mary Hudachek-Buswell. "Studies on the effectiveness of virtual pointers in collaborative augmented reality". In: *2008 IEEE Symposium on 3D User Interfaces*. IEEE. 2008, pp. 117–124.

[18] Sicheng Chen, Miao Chen, Andreas Kunz, Asim Evren Yantaç, Mathias Bergmark, Anders Sundin, and Morten Fjeld. "SEMarbeta: mobile sketch-gesture-video remote support for car drivers". In: *Proceedings of the 4th Augmented Human International Conference*. ACM. 2013, pp. 69–76.

[19] Yang Chen and Gérard Medioni. "Object modelling by registration of multiple range images". In: *Image and vision computing* 10.3 (1992), pp. 145–155.

[20] Mauro Cherubini, Marc-Antoine Nüssli, and Pierre Dillenbourg. "Deixis and gaze in collaborative work at a distance (over a shared map): a computational model to detect misunderstandings". In: *Proceedings of the 2008 symposium on Eye tracking research & applications*. ACM. 2008, pp. 173–180.

[21] François Coldefy and Stphane Louis-dit-Picard. "DigiTable: an interactive multiuser table for collocated and remote collaboration enabling remote gesture

visualization". In: *2007 IEEE conference on computer vision and pattern recognition*. IEEE. 2007, pp. 1–8.

[22] Dragoş Datcu, Stephan G. Lukosch, and Heide K. Lukosch. "Handheld Augmented Reality for Distributed Collaborative Crime Scene Investigation". In: *Proceedings of the 19th International Conference on Supporting Group Work - GROUP '16* (2016), pp. 267–276.

[23] Dragoş Datcu, Thomas Swart, Stephan Lukosch, and Zoltan Rusak. "Multimodal collaboration for crime scene investigation in mediated reality". In: *Proceedings of the 14th ACM international conference on Multimodal interaction*. ACM. 2012, pp. 299–300.

[24] Mingsong Dou, Henry Fuchs, and Jan-Michael Frahm. "Scanning and tracking dynamic objects with commodity depth cameras". In: *2013 IEEE international symposium on mixed and augmented Reality (ISMAR)*. IEEE. 2013, pp. 99–106.

[25] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. "Fusion4d: Real-time performance capture of challenging scenes". In: *ACM Transactions on Graphics (TOG)* 35.4 (2016), pp. 1–13.

[26] Carmine Elvezio, Mengu Sukan, Ohan Oda, Steven Feiner, and Barbara Tversky. "Remote collaboration in AR and VR using virtual replicas". In: *ACM SIGGRAPH 2017 VR Village* (2017), p. 13.

[27] Katherine M Everitt, Scott R Klemmer, Robert Lee, and James A Landay. "Two worlds apart: bridging the gap between physical and virtual media for distributed design collaboration". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2003, pp. 553–560.

[28] Ching-Ling Fan, Jean Lee, Wen-Chih Lo, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. "Fixation prediction for 360 video streaming in head-mounted virtual reality". In: *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video* (2017), pp. 67–72.

[29] Susan R Fussell, Robert E Kraut, and Jane Siegel. "Coordination of communication: Effects of shared visual context on collaborative work". In: *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. ACM. 2000, pp. 21–30.

[30] Susan R Fussell, Leslie D Setlock, and Robert E Kraut. "Effects of head-mounted and scene-oriented video systems on remote collaboration on physical tasks". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2003, pp. 513–520.

[31] Susan R Fussell, Leslie D Setlock, Jie Yang, Jiazhi Ou, Elizabeth Mauer, and Adam DI Kramer. "Gestures over video streams to support remote collaboration on physical tasks". In: *Human-Computer Interaction* 19.3 (2004), pp. 273–309.

[32] Lei Gao, Huidong Bai, Mark Billinghurst, and Robert W Lindeman. "User Behaviour Analysis of Mixed Reality Remote Collaboration with a Hybrid View Interface". In: *32nd Australian Conference on Human-Computer Interaction*. 2020, pp. 629–638.

[33] Lei Gao, Huidong Bai, Weiping He, Mark Billinghurst, and Robert W Lindeman. "Real-time visual representations for mobile mixed reality remote collaboration". In: *SIGGRAPH Asia 2018 Virtual & Augmented Reality*. ACM. 2018, p. 15.

[34] Lei Gao, Huidong Bai, Gun Lee, and Mark Billinghurst. "An oriented point-cloud view for MR remote collaboration". In: *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications*. ACM. 2016, 8:1–8:4.

[35] Lei Gao, Huidong Bai, Robert W Lindeman, and Mark Billinghurst. "Static local environment capturing and sharing for MR remote collaboration". In: *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*. ACM. 2017, p. 17.

[36] Lei Gao, Huidong Bai, Thammathip Piumsomboon, Gun A Lee, Robert W Lindeman, and Mark Billinghurst. "Real-time visual representations for mixed reality remote collaboration". In: *Proceedings of the 27th International Conference on Artificial Reality and Telexistence and 22nd Eurographics Symposium on Virtual Environments*. Eurographics Association. 2017, pp. 87–95.

[37] Steffen Gauglitz, Cha Lee, Matthew Turk, and Tobias Höllerer. "Integrating the physical environment into mobile remote collaboration". In: *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*. ACM. 2012, pp. 241–250.

[38] Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. "In touch with the remote world: Remote collaboration with augmented reality drawings and virtual navigation". In: *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology* (2014), pp. 197–205.

[39] Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. "World-stabilized annotations and virtual scene navigation for remote collaboration". In: *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM. 2014, pp. 449–459.

[40] William W Gaver, Abigail Sellen, Christian Heath, and Paul Luff. "One is not enough: Multiple views in a media space". In: *Proceedings of the INTERACT'93 and*

CHI'93 Conference on Human Factors in Computing Systems. ACM. 1993, pp. 335–341.

[41] *Global 360-Degree Camera Market 2016-2020*. 2016. URL: https://goo.gl/zJCdnO.

[42] Giovani Gomez. "On selecting colour components for skin detection". In: *Object recognition supported by user interaction for service robots*. Vol. 2. IEEE. 2002, pp. 961–964.

[43] Markus Gross, Stephan Würmlin, Martin Naef, Edouard Lamboray, Christian Spagno, Andreas Kunz, Esther Koller-Meier, Tomas Svoboda, Luc Van Gool, Silke Lang, et al. "blue-c: a spatially immersive display and 3D video portal for telepresence". In: *ACM Transactions on Graphics (TOG)* 22.3 (2003), pp. 819–827.

[44] Jan Gugenheimer, Dennis Wolf, Gabriel Haas, Sebastian Krebs, and Enrico Rukzio. "Swivrchair: A motorized swivel chair to nudge users' orientation for 360 degree storytelling in virtual reality". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016), pp. 1996–2000.

[45] Ravindra Guntur and Wei Tsang Ooi. "On tile assignment for region-of-interest video streaming in a wireless LAN". In: *Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video* (2012), pp. 59–64.

[46] Kunal Gupta, Gun A Lee, and Mark Billinghurst. "Do you see what I see? The effect of gaze tracking on task space remote collaboration". In: *IEEE transactions on visualization and computer graphics* 22.11 (2016), pp. 2413–2422.

[47] Pavel Gurevich, Joel Lanir, Benjamin Cohen, and Ran Stone. "TeleAdvisor: a versatile augmented reality tool for remote assistance". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2012, pp. 619–622.

[48] Chad Harms and Frank Biocca. "Internal Consistency and Reliability of the Networked Minds Measure of Social Presence". In: *Seventh Annual International Workshop: Presence 2004* (2004), pp. 246–251.

[49] Keita Higuch, Ryo Yonetani, and Yoichi Sato. "Can Eye Help You?: Effects of Visualizing Eye Fixations on Remote Collaboration Scenarios for Physical Tasks". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16 (2016), pp. 5180–5190.

[50] Weidong Huang and Leila Alem. "Supporting hand gestures in mobile remote collaboration: a usability evaluation". In: *Proceedings of the 25th BCS Conference on Human-Computer Interaction*. British Computer Society. 2011, pp. 211–216.

[51] Hiroshi Ishii, Minoru Kobayashi, and Kazuho Arita. "Interactive design of seamless collaboration media". In: *Communications of the ACM* 37.8 (1994), pp. 83–98.

[52] Hiroshi Ishii, Minoru Kobayashi, and Jonathan Grudin. "Integration of interpersonal space and shared workspace: ClearBoard design and experiments". In: *ACM Transactions on Information Systems (TOIS)* 11.4 (1993), pp. 349–375.

[53] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera". In: *Proceedings of the 24th annual ACM symposium on User interface software and technology* (2011), pp. 559–568.

[54] P. Jermann, D. Gergle, R. Bednarik, and P. Dillenbourg. "DUET 2013: Dual eye tracking in CSCL". In: *Computer-Supported Collaborative Learning Conference, CSCL* 2 (2013), pp. 446–451.

[55] Steven Johnson, Madeleine Gibson, and Bilge Mutlu. "Handheld or handsfree?: Remote collaboration via lightweight head-mounted displays and handheld devices". In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing* (2015), pp. 1825–1836.

[56] Olaf Kähler, Victor A Prisacariu, and David W Murray. "Real-time large-scale dense 3D reconstruction with loop closure". In: *European Conference on Computer Vision* (2016), pp. 500–516.

[57] Olaf Kähler, Victor Adrian Prisacariu, Carl Yuheng Ren, Xin Sun, Philip Torr, and David Murray. "Very high frame rate volumetric integration of depth images on mobile devices". In: *IEEE transactions on visualization and computer graphics* 21.11 (2015), pp. 1241–1250.

[58] Shunichi Kasahara, Shohei Nagai, and Jun Rekimoto. "LiveSphere: Immersive Experience Sharing with 360 degrees Head-mounted Cameras". In: *Proceedings of the adjunct publication of the 27th annual ACM symposium on User interface software and technology - UIST'14 Adjunct* (2014), pp. 61–62.

[59] Robert S Kennedy, Norman E Lane, Kevin S Berbaum, and Michael G Lilienthal. "Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness". In: *The international journal of aviation psychology* 3.3 (1993), pp. 203–220.

[60] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. "Real time hand pose estimation using depth sensors". In: *Consumer depth cameras for computer vision*. Springer, 2013, pp. 119–137.

[61] Seungwon Kim, Gun Lee, Nobuchika Sakata, and Mark Billinghurst. "Improving co-presence with augmented visual communication cues for sharing experience through video conference". In: *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2014, pp. 83–92.

[62] Seungwon Kim, Gun A Lee, Nobuchika Sakata, Andreas Dünser, Elina Vartiainen, and Mark Billinghurst. "Study of augmented gesture communication cues and view sharing in remote collaboration". In: *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2013, pp. 261–262.

[63] Hideaki Kimata, Daisuke Ochi, Akio Kameda, Hajime Noto, Katsuhiko Fukazawa, and Akira Kojima. "Mobile and multi-device interactive panorama video distribution system". In: *The 1st IEEE Global Conference on Consumer Electronics 2012* (2012), pp. 574–578.

[64] David Kirk, Tom Rodden, and Danaë Stanton Fraser. "Turn it this way: grounding collaborative action with remote gestures". In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM. 2007, pp. 1039–1048.

[65] David Kirk and Danae Stanton Fraser. "Comparing remote gesture technologies for supporting collaborative physical tasks". In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM. 2006, pp. 1191–1200.

[66] Alexandros Kitsikidis, Kosmas Dimitropoulos, Stella Douka, and Nikos Grammalidis. "Dance analysis using multiple kinect sensors". In: *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*. Vol. 2. IEEE. 2014, pp. 789–795.

[67] Nikolina Koleva, Sabrina Hoppe, Mohammad Mehdi Moniri, Maria Staudte, and Andreas Bulling. "On the interplay between spontaneous spoken instructions and human visual behaviour in an indoor guidance task." In: *CogSci*. 2015.

[68] Jure Kovac, Peter Peer, and Franc Solina. "Human skin colour clustering for face detection". In: *IEEE Region 8 EUROCON 2003: Computer as a Tool - Proceedings* B (2003), pp. 144–148.

[69] Marek Kowalski, Jacek Naruniec, and Michal Daniluk. "Livescan3d: A fast and inexpensive 3d data acquisition system for multiple kinect v2 sensors". In: *2015 International Conference on 3D Vision*. IEEE. 2015, pp. 318–325.

[70] Takeshi Kurata, Nobuchika Sakata, Masakatsu Kourogi, Hideaki Kuzuoka, and Mark Billinghurst. "Remote collaboration using a shoulder-worn active camera/laser". In: *Eighth international symposium on wearable computers*. Vol. 1. IEEE. 2004, pp. 62–69.

[71] Joel Lanir, Ran Stone, Benjamin Cohen, and Pavel Gurevich. "Ownership and control of point of view in remote assistance". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2013, pp. 2243–2252.

[72] Morgan Le Chénéchal, Thierry Duval, Valérie Gouranton, Jérôme Royan, and Bruno Arnaldi. "Vishnu: virtual immersive support for HelpiNg users an interaction paradigm for collaborative remote guiding in mixed reality". In: *2016 IEEE Third VR International Workshop on Collaborative Virtual Environments (3DCVE)* (2016), pp. 9–12.

[73] Gun A Lee, Theophilus Teo, Seungwon Kim, and Mark Billinghurst. "A user study on MR remote collaboration using live 360 video". In: *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2018), pp. 153–164.

[74] Gun A Lee, Theophilus Teo, Seungwon Kim, and Mark Billinghurst. "Mixed reality collaboration through sharing a live panorama". In: *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications* (2017), p. 14.

[75] Gun A Lee, Theophilus Teo, Seungwon Kim, and Mark Billinghurst. "Sharedsphere: MR collaboration through shared live panorama". In: *SIGGRAPH Asia 2017 Emerging Technologies* (2017), p. 12.

[76] WonSook Lee, Jin Gu, and Nadia Magnenat-Thalmann. "Generating animatable 3D virtual humans from photographs". In: *Computer Graphics Forum*. Vol. 19. 3. Wiley Online Library. 2000, pp. 1–10.

[77] Bo Li, Lionel Heng, Kevin Koser, and Marc Pollefeys. "A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 1301–1307.

[78] Jerry Li, Mia Manavalan, Sarah D'Angelo, and Darren Gergle. "Designing shared gaze awareness for remote collaboration". In: *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion*. ACM. 2016, pp. 325–328.

[79] Yen-Chen Lin, Yung-Ju Chang, Hou-Ning Hu, Hsien-Tzu Cheng, Chi-Wen Huang, and Min Sun. "Tell me where to look: Investigating ways for assisting focus in 360 video". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (2017), pp. 2535–2545.

[80] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[81] Aditya Mavlankar and Bernd Girod. "Pre-fetching based on video analysis for interactive region-of-interest streaming of soccer sequences". In: *2009 16th IEEE International Conference on Image Processing (ICIP)* (2009), pp. 3061–3064.

[82] Aditya Mavlankar and Bernd Girod. "Video streaming with interactive pan/t-ilt/zoom". In: *High-Quality Visual Experience* (2010), pp. 431–455.

[83] Paul Milgram and Fumio Kishino. "A taxonomy of mixed reality visual displays". In: *IEICE TRANSACTIONS on Information and Systems* 77.12 (1994), pp. 1321–1329.

[84] Serguei A Mokhov, Miao Song, Jonathan Llewellyn, Jie Zhang, Alexander Charette, Ruofan Wu, and Shuiying Ge. "Real-time collection and analysis of 3-Kinect v2 skeleton data in a single application". In: *ACM SIGGRAPH 2016 Posters*. ACM. 2016, p. 53.

[85] Björn Müller, Winfried Ilg, Martin A Giese, and Nicolas Ludolph. "Validation of enhanced kinect sensor based motion capturing for gait assessment." In: *PLoS ONE* 12.4 (2017).

[86] Romy Müller, Jens R Helmert, and Sebastian Pannasch. "Limitations of gaze transfer: Without visual context, eye movements do not to help to coordinate joint action, whereas mouse movements do". In: *Acta psychologica* 152 (2014), pp. 19–28.

[87] Romy Müller, Jens R Helmert, Sebastian Pannasch, and Boris M Velichkovsky. "Gaze transfer in remote cooperation: Is it always helpful to see what your partner is attending to?" In: *The Quarterly Journal of Experimental Psychology* 66.7 (2013), pp. 1302–1316.

[88] David T Nguyen and John Canny. "Multiview: improving trust in group video conferencing through spatial faithfulness". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2007, pp. 1465–1474.

[89] Ohan Oda, Carmine Elvezio, Mengu Sukan, Steven Feiner, and Barbara Tversky. "Virtual replicas for remote assistance in virtual and augmented reality". In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (2015), pp. 405–415.

[90] Ohan Oda, Mengu Sukan, Steven Feiner, and Barbara Tversky. "Poster: 3D referencing for remote task assistance in augmented reality". In: *2013 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE. 2013, pp. 179–180.

[91] Okreylos. *Lighthouse tracking examined*. 2016. URL: http://doc-ok.org/?p=1478.

[92] Nuria Oliver, Alex Pentland, and François Bérard. "LAFTER: a real-time face and lips tracker with facial expression recognition". In: *Pattern Recognition* 33.8 (2000), pp. 1369–1382. ISSN: 0031-3203.

[93] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al. "Holoportation: Virtual 3d teleportation in real-time". In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 2016, pp. 741–754.

[94] Nobuyuki Otsu. "A threshold selection method from gray-level histograms". In: *IEEE transactions on systems, man, and cybernetics* 9.1 (1979), pp. 62–66.

[95] Ronald Poelman, Oytun Akman, Stephan Lukosch, and Pieter Jonker. "As if being there: mediated reality for crime scene investigation". In: *Proceedings of the ACM 2012 conference on computer supported cooperative work*. ACM. 2012, pp. 1267–1276.

[96] Victor Adrian Prisacariu, Olaf Kähler, Stuart Golodetz, Michael Sapienza, Tommaso Cavallari, Philip HS Torr, and David W Murray. "InfiniTAM v3: a framework for large-scale 3D reconstruction with loop closure". In: *arXiv preprint arXiv:1708.00783* (2017).

[97] Abhishek Ranjan, Jeremy P Birnholtz, and Ravin Balakrishnan. "Dynamic shared visual spaces: experimenting with automatic camera control in a remote repair task". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2007, pp. 1177–1186.

[98] Yvonne Rogers and Judi Ellis. "Distributed cognition: an alternative framework for analysing and explaining collaborative working". In: *Journal of information technology* 9.2 (1994), pp. 119–128.

[99] Henry Roth and Marsette Vona. "Moving Volume KinectFusion". In: *Procedings of the British Machine Vision Conference 2012* (2012), pp. 112.1–112.11.

[100] Gavriel Salomon. *Distributed cognitions: Psychological and educational considerations*. Cambridge University Press, 1997.

[101] Prasanth Sasikumar, Lei Gao, Huidong Bai, and Mark Billinghurst. "Wearable RemoteFusion: A Mixed Reality Remote Collaboration System with Local Eye Gaze and Remote Hand Gesture Sharing". In: *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE. 2019, pp. 393–394.

[102] Jeff Sauro. "Measuring usability with the system usability scale". In: *URL: http://-www. measuringusability. com/sus. php and procedure (9.4. 2012.)* (2011).

[103] Jeff Sauro and Joseph S. Dumas. "Comparison of three one-question, post-task usability questionnaires". In: *Conference on Human Factors in Computing Systems - Proceedings* April 2009 (2009), pp. 1599–1608.

[104] Bertrand Schneider and Roy Pea. "Real-time mutual gaze perception enhances collaborative learning and collaboration quality". In: *International Journal of Computer-supported collaborative learning* 8.4 (2013), pp. 375–397.

[105] Dongmahn Seo, Suhyun Kim, JaeWook Yoo, Hogun Park, and Heedong Ko. "Immersive panorama TV service system". In: *2012 IEEE international conference on consumer electronics (ICCE)* (2012), pp. 201–202.

[106] Rodrigo Silva, Bruno Feijó, Pablo B Gomes, Thiago Frensh, and Daniel Monteiro. "Real time 360° video stitching and streaming". In: *ACM SIGGRAPH 2016 Posters*. ACM. 2016, p. 70.

[107] Rajinder S Sodhi, Brett R Jones, David Forsyth, Brian P Bailey, and Giuliano Maciocci. "BeThere: 3D mobile collaboration with spatial input". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2013, pp. 179–188.

[108] Son Lam Phung, A. Bouzerdoum, and D. Chai. "A novel skin color model in YCbCr color space and its application to human face detection". In: *Proceedings. International Conference on Image Processing* 1 (2002), pp. I–289–I–292.

[109] Aaron Stafford and Wayne Piekarski. "User evaluation of god-like interaction techniques". In: *Proceedings of the ninth conference on Australasian user interface-Volume 76* (2008), pp. 19–27.

[110] Basu Prasad Subedi. "Using Likert type data in social science research: Confusion, issues and challenges". In: *International journal of contemporary applied sciences* 3.2 (2016), pp. 36–49.

[111] Matthew Tait and Mark Billinghurst. "The effect of view independence in a collaborative AR system". In: *Computer Supported Cooperative Work (CSCW)* 24.6 (2015), pp. 563–589.

[112] Franco Tecchia, Leila Alem, and Weidong Huang. "3D helping hands: a gesture based MR system for remote collaboration". In: *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*. ACM. 2012, pp. 323–328.

[113] Theophilus Teo, Louise Lawrence, Gun A. Lee, Mark Billinghurst, and Matt Adcock. "Mixed reality remote collaboration combining 360 video and 3D reconstruction". In: *Conference on Human Factors in Computing Systems - Proceedings* (2019), pp. 1–14.

[114] Nadia Magnenat Thalmann, Zerrin Yumak, and Aryel Beck. "Autonomous virtual humans and social robots in telepresence". In: *2014 IEEE 16th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE. 2014, pp. 1–6.

[115] Sebastian Thrun and John J Leonard. "Simultaneous localization and mapping". In: *Springer handbook of robotics* (2008), pp. 871–889.

[116] Brygg Ullmer and Hiroshi Ishii. "The metaDESK: models and prototypes for tangible user interfaces". In: *Proceedings of the 10th annual ACM symposium on User interface software and technology - UIST '97* (1997), pp. 223–232.

[117] Brygg Ullmer, Hiroshi Ishii, and Dylan Glas. "mediaBlocks: physical containers, transports, and controls for online media". In: *Proceedings of SIGGRAPH'98*. 1998.

[118] John Underkoffler and Hiroshi Ishii. "Illuminating Light: An Optical Design Tool with a Luminous-tangible Interface". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '98 (1998), pp. 542–549.

[119] Vladimir Vezhnevets, Vassili Sazonov, and Alla Andreeva. "A survey on pixel-based skin color detection techniques". In: *Proc. Graphicon*. Vol. 3. Moscow, Russia. 2003, pp. 85–92.

[120] Peter Vorderer, Werner Wirth, Feliz Ribeiro Gouveia, Frank Biocca, Timo Saari, Lutz Jäncke, Saskia Böcking, Holger Schramm, Andre Gysbers, Tilo Hartmann, et al. "Mec spatial presence questionnaire". In: *Retrieved Sept* 18 (2004), p. 2015.

[121] Xiangyu Wang, Peter ED Love, Mi Jeong Kim, and Wei Wang. "Mutual awareness in collaborative design: An Augmented Reality integrated telepresence system". In: *Computers in Industry* 65.2 (2014), pp. 314–324.

[122] Yuanjie Wu, Lei Gao, Simon Hoermann, and Robert W Lindeman. "Towards Robust 3D Skeleton Tracking Using Data Fusion from Multiple Depth Sensors". In: *2018 10th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)* (2018), pp. 1–4.

[123] Hiroyuki Yabe, Daichi Ono, and Tsutomu Horikawa. "Space fusion: context-aware interaction using 3D scene parsing". In: *SIGGRAPH Asia 2018 Virtual & Augmented Reality*. ACM. 2018, p. 17.

[124] Ming Zeng, Fukai Zhao, Jiaxiang Zheng, and Xinguo Liu. "A memory-efficient kinectfusion using octree". In: *International Conference on Computational Visual Media*. Springer. 2012, pp. 234–241.

Appendix A

Questionnaires

This section presents the questionnaire used in the user study presented in Chapter 7 part two, in which our proposed MR remote collaboration system combined three types of views (a static view, a live 2D first-person view and a live 360° God-like view) for users to chose based on their own choices.

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

What is your gender?

- ◯ Male
- ◯ Female
- ◯ Other
- ◯ Prefer not to disclose

How old are you?

Have you used any augmented reality or virtual reality applications before?

- ◯ Not at all
- ◯ Once
- ◯ Daily
- ◯ Few times a week
- ◯ Few times a month
- ◯ Few times a year

How often do you use live video streaming apps?

- ◯ Not at all (Please skip next question)
- ◯ Once
- ◯ Daily
- ◯ Few times a week

- ◯ Few times a month
- ◯ Few times a year

Which app do you use for live video streaming?

- ☐ Skype
- ☐ Facebook live
- ☐ Snapchat
- ☐ Periscope
- ☐ WeChat
- ☐ Other

What is your main purpose for using live video streaming?

- ☐ Social connection with friends
- ☐ Connection with family members
- ☐ Commercial or business meeting
- ☐ Other

→

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Simulator sickness questionnaire (SSQ):

None Severe
0 1 2 3 4 5 6 7 8 9

General discomfort

Fatigue

Headache

Eyestrain

Difficulty fousing

Increased salivation

Sweating

Nausea

Difficulty concentrating

Vertigo

Blurred vision

Dizziness (eyss open)

Dizziness (eyes closed)

Stomach awareness

→

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Single Ease Question (SEQ)

	Very difficult			medium			very easy
(1) Overall, the task was…	◯	◯	◯	◯	◯	◯	◯

Custom user experience rating questions

	strongly disagree	Disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Agree	strongly agree
(1) I enjoyed the experience	◯	◯	◯	◯	◯	◯	◯
(2) I was able to focus on the task	◯	◯	◯	◯	◯	◯	◯
(3) I understood where my partner's focus was on	◯	◯	◯	◯	◯	◯	◯

→

Powered by Qualtrics

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

The social presence questionnaire (SoPQ)

	strongly disagree	Disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Agree	Strongly agree
(1) I noticed my partner	○	○	○	○	○	○	○
(2) My partner noticed me	○	○	○	○	○	○	○
(3) My partner's presence was obvious to me	○	○	○	○	○	○	○
(4) My presence was obvious to my partner	○	○	○	○	○	○	○
(5) My partner caught my attention	○	○	○	○	○	○	○
(6) I caught my partner's attention	○	○	○	○	○	○	○
(7) My thoughts were clear to my partner	○	○	○	○	○	○	○
(8) My partner's thoughts were clear to me	○	○	○	○	○	○	○
(9) It was easy to understand my partner	○	○	○	○	○	○	○

	strongly disagree	Disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Agree	Strongly agree
(10) My partner found it easy to understand me	○	○	○	○	○	○	○
(11) Understanding my partner was difficult	○	○	○	○	○	○	○
(12) My partner had difficulty understanding me	○	○	○	○	○	○	○
(13) My behavior was often in direct response to my partner's behavior	○	○	○	○	○	○	○
(14) The behavior of my partner was often in direct response to my behavior	○	○	○	○	○	○	○
(15) I reciprocated my partner's actions	○	○	○	○	○	○	○
(16) My partner reciprocated my actions	○	○	○	○	○	○	○
(17) My partner's behavior was closely tied to my behavior.	○	○	○	○	○	○	○

	strongly disagree	Disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Agree	Strongly agree
(18) My behavior was closely tied to my partner's behavior	○	○	○	○	○	○	○
(19) I was easily distracted from my partner when other things were going on	○	○	○	○	○	○	○
(20) My partner was easily distracted from me when other things were going on	○	○	○	○	○	○	○
(21) I remained focused on (my partner) throughout our interaction	○	○	○	○	○	○	○
(22) My partner remained focused on me throughout our interaction.	○	○	○	○	○	○	○
(23) My partner did not receive my full attention	○	○	○	○	○	○	○
(24) I did not receive my partner's full attention	○	○	○	○	○	○	○

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

MEC spatial presence questionnaire (SpPQ) : Spatial presence self-location (SPSL)

	fully disagree		Neither agree nor disagree		fully agree
(1) I felt like I was actually there in the environment of the presentation	○	○	○	○	○
(2) It was as though my true location had shifted into the environment in the presentation.	○	○	○	○	○
(3) I felt as though I was physically present in the environment of the presentation.	○	○	○	○	○
(4) It seemed as though I actually took part in the action of the presentation.	○	○	○	○	○

MEC spatial presence questionnaire (SpPQ) : Spatial situation model (SSM)

	fully disagree		Neither agree nor disagree		fully agree
(1) I was able to imagine the arrangement of the spaces presented in the local side very well.	○	○	○	○	○

	fully disagree		Neither agree nor disagree		fully agree
(2) I had a precise idea of the spatial surroundings presented in the local side.	○	○	○	○	○
(3) I was able to make a good estimate of the size of the presented space.	○	○	○	○	○
(4) Even now, I still have a concrete mental image of the spatial environment.	○	○	○	○	○

→

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

System usability scale (SUS)

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
(1) I think that I would like to use this system frequently	○	○	○	○	○
(2) I found the system unnecessarily complex	○	○	○	○	○
(3) I thought the system was easy to use	○	○	○	○	○
(4) I think that I would need the support of a technical person to be able to use this system	○	○	○	○	○
(5) I found the various functions in this system were well integrated	○	○	○	○	○
(6) I thought there was too much inconsistency in this system	○	○	○	○	○
(7) I would imagine that most people would learn to use this system very quickly	○	○	○	○	○

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
(8) I found the system very cumbersome to use	○	○	○	○	○
(9) I felt very confident using the system	○	○	○	○	○
(10) I needed to learn a lot of things before I could get going with this system	○	○	○	○	○

→

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Please rank (1:best, 2:medium, 3:worst) the three view modes based on the following categories:

(1) Communicating with partner

- 3D static global view
- 2D live first-person view
- 2D live 360 God-view

(2) Understanding the partner's focus

- 3D static global view
- 2D live first-person view
- 2D live 360 God-view

(3) Guiding

- 3D static global view
- 2D live first-person view
- 2D live 360 God-view

(4) Item searching

3D static global view

2D live first-person view

2D live 360 God-view

View switching

	Strongly disagree		Neither agree nor disagree		Strongly agree
(1) I think the view switching between different view modes is easy to use	○	○	○	○	○
(2) I think the view switching between different view modes is smooth	○	○	○	○	○
(3) I think using this view switching is comfortable	○	○	○	○	○

What's the advantages of this remote collaboration system?

What's the limitations of this remote collaboration system?

What's your suggestion about this remote collaboration system?

→

Appendix B

Co-authorship Form

This form is to accompany the submission of any thesis that contains research reported in co-authored work that has been published, accepted for publication, or submitted for publication.

Deputy Vice-Chancellor's Office
Postgraduate Research Office

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Co-Authorship Form

This form is to accompany the submission of any thesis that contains research reported in co-authored work that has been published, accepted for publication, or submitted for publication. A copy of this form should be included for each co-authored work that is included in the thesis. Completed forms should be included at the front (after the thesis abstract) of each copy of the thesis submitted for examination and library deposit.

Please indicate the chapter/section/pages of this thesis that are extracted from co-authored work and provide details of the publication or submission from the extract comes:

Chapter 3*:* Gao, L., Bai, H., Lee, G., & Billinghurst, M. (2016, November). An oriented point-cloud view for MR remote collaboration. In *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications* (p. 8). ACM.

Chapter 4: Gao, L., Bai, H., Piumsomboon, T., Lee, G., Lindeman, R. W., & Billinghurst, M. (2017). Real-time visual representations for mixed reality remote collaboration.

Chapter 5: Gao, L., Bai, H., Lindeman, R., & Billinghurst, M. (2017, November). Static local environment capturing and sharing for MR remote collaboration. In *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications* (p. 17). ACM.

Chapter 6: Gao, L., Bai, H., Piumsomboon, T., Lee, G., Lindeman, R. W., & Billinghurst, M. (2017). Real-time visual representations for mixed reality remote collaboration.

Chapter 8: Gao, L., Bai, H., He, W., Billinghurst, M., & Lindeman, R. W. (2018, December). Real-time visual representations for mobile mixed reality remote collaboration. In *SIGGRAPH Asia 2018 Virtual & Augmented Reality* (p. 15). ACM.

Please detail the nature and extent (%) of contribution by the candidate:
The author of this PhD thesis was the main contributor at all stages of the work including the system development, interface design, user study design, running the

user studies, data analysis, final evaluations of the studies and paper writing for all the publications listed above, which took around 80% of the contribution. Dr. Huidong Bai was mainly involved in the interface design, user study design and thesis writing for the publications listed above. The other co-authors were mainly involved in interface and user study design for the publications listed above.

Certification by Co-authors:
If there is more than one co-author then a single co-author can sign on behalf of all
The undersigned certifies that:

- The above statement correctly reflects the nature and extent of the PhD candidate's contribution to this co-authored work
- In cases where the candidate was the lead author of the co-authored work he or she wrote the text

Name: *Huidong Bai* Signature:  Date: *26/-7/2019*